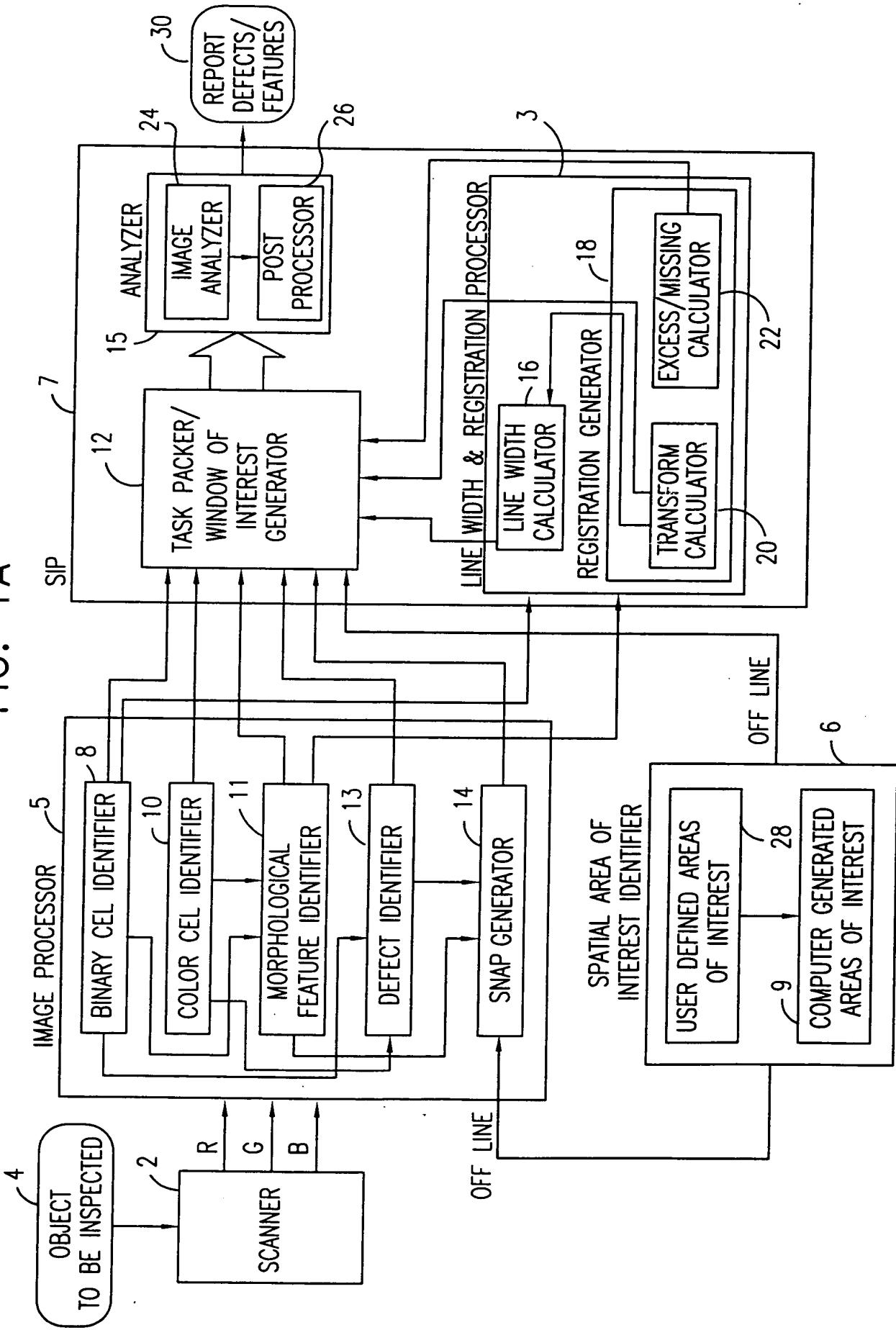


FIG. 1 A



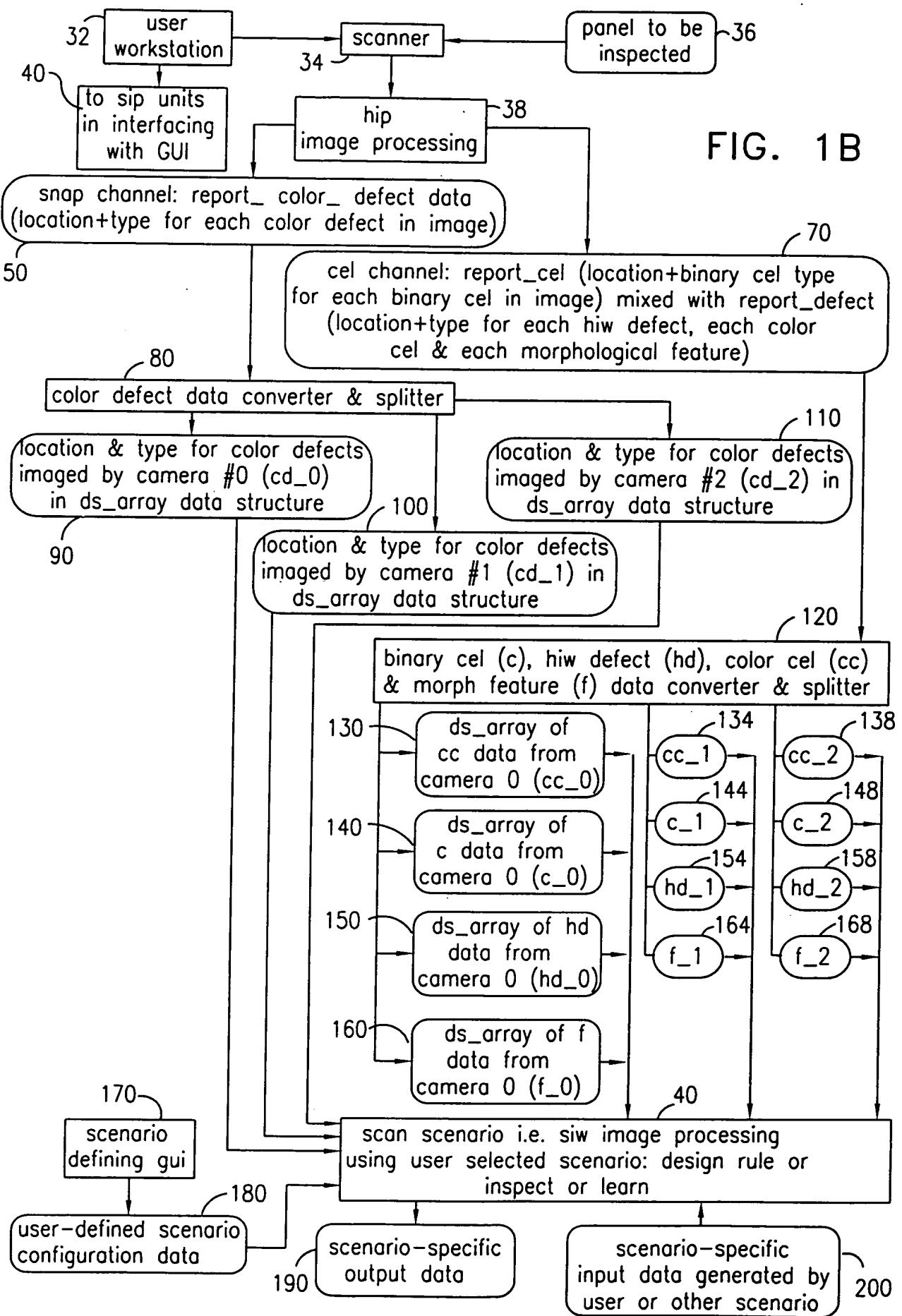


FIG. 2

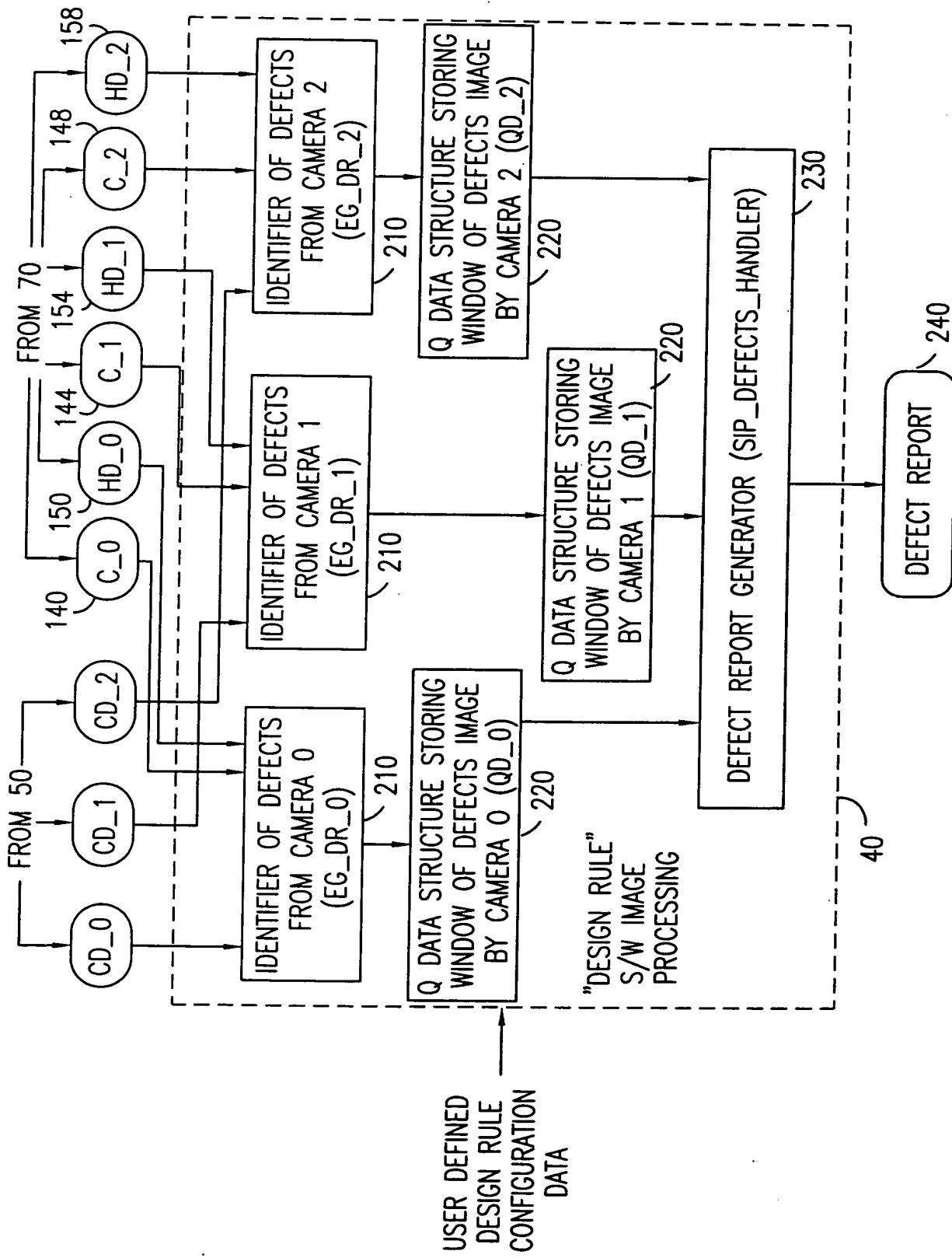
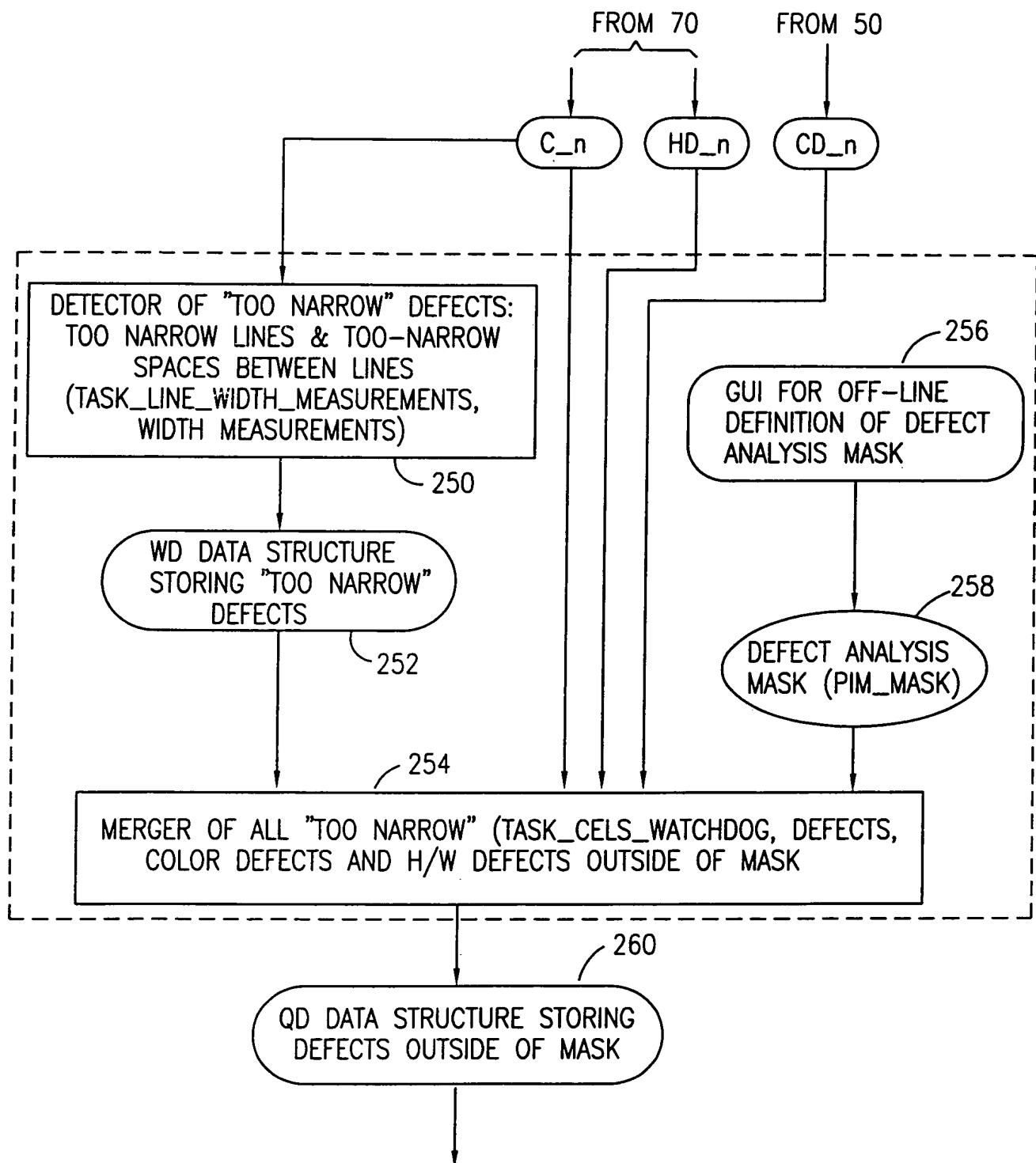
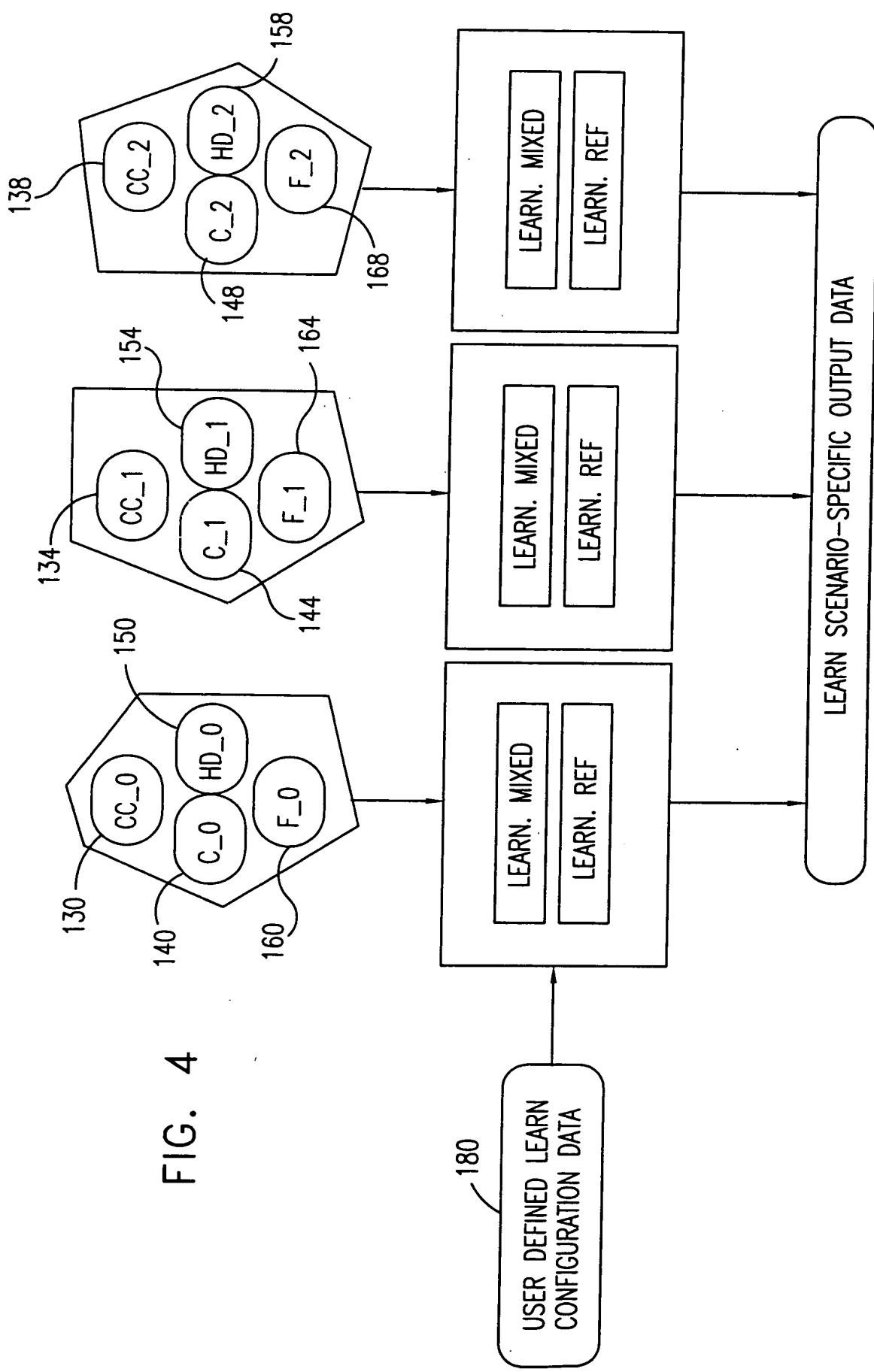


FIG. 3





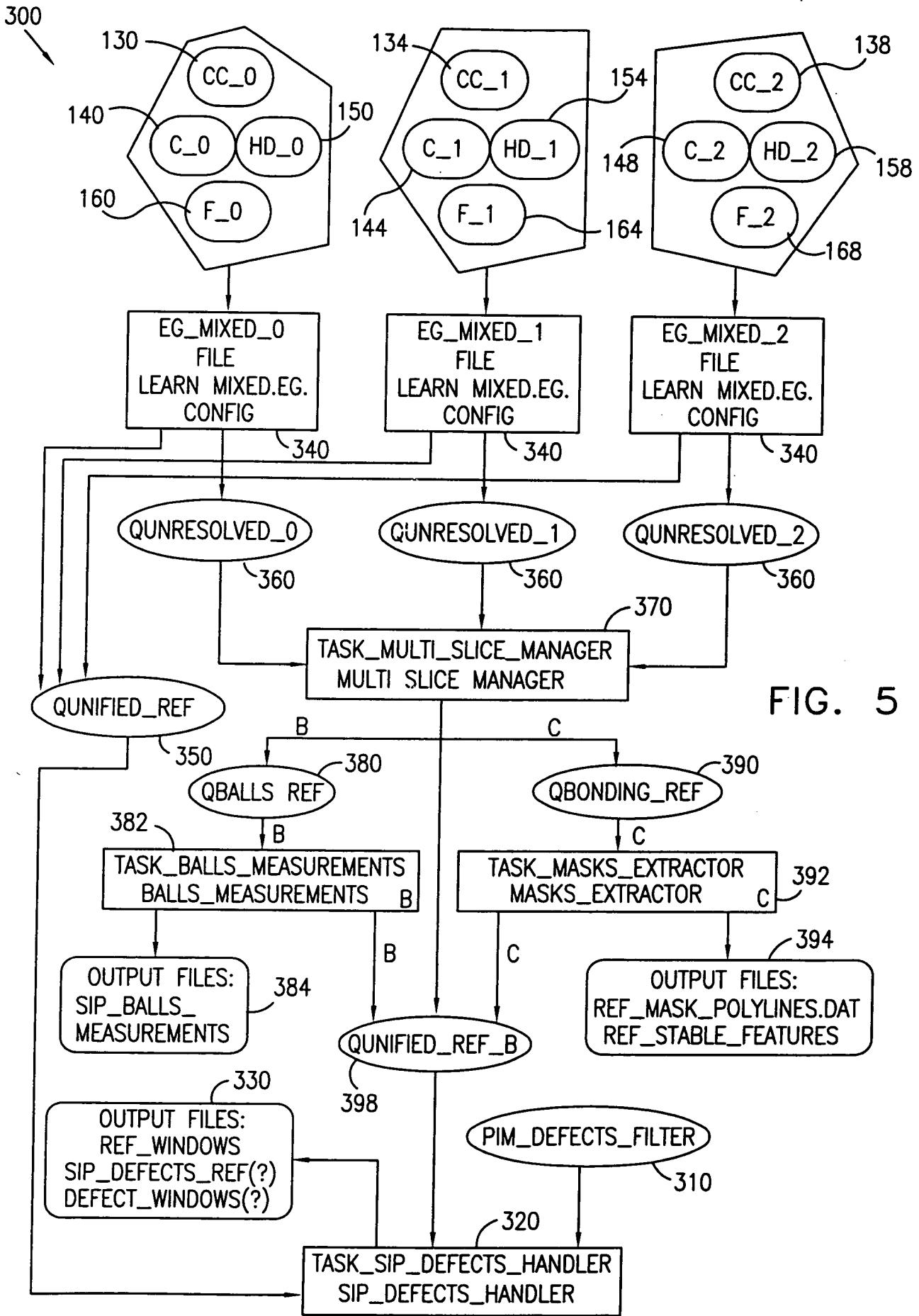


FIG. 5

FIG. 6

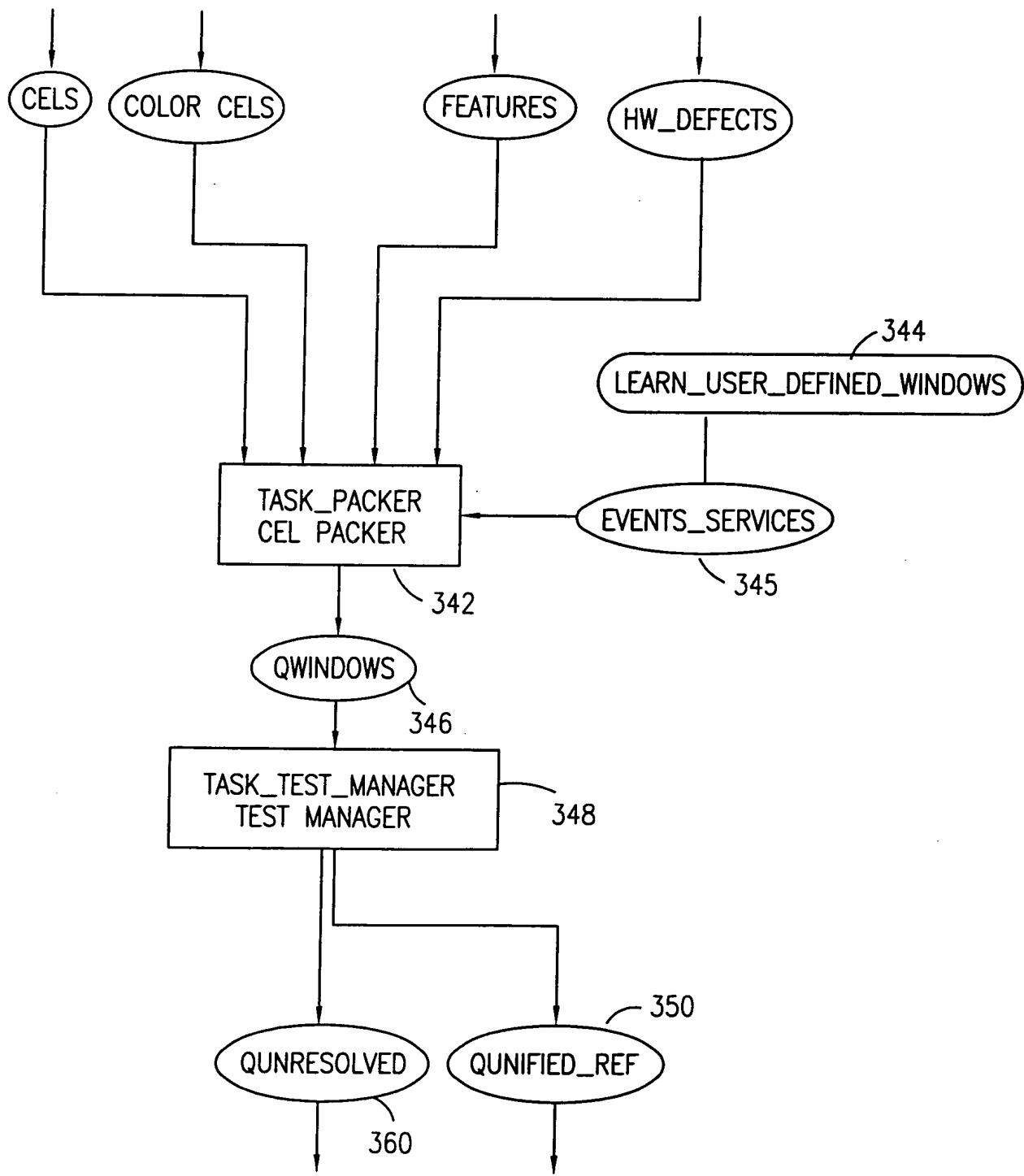


FIG. 7A

Window type	Functions attached	Remarks
target	<u>target_lrn:</u> connect_closed (Func_connected_components<CEL>) vectorize (Func_angle_vectorizer<Cel>) target_reference (currently does nothing) trans2ref (Func_trans2ref) forward2target_analyze (Func_forward)	Produces connected components of raw CELs. Vectorizes them into polylines. Then transforms vectorized polylines into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is target_analyze.
bonding_area	ii_lrn <u>connect_closed</u> (Func_connected_components<CEL>) vectorize (Func_angle_vectorizer<CEL>) connect_open_color (only in enable_color_masking mode) (Func_connected_components<Color_cel>) vectorize_color (only in enable_color_masking mode) (Func_angle_vectorizer_<Color_cel>) stp_reference trans2ref (Func_trans2ref) forward2bonding_analyze (Func_forward)	Produces connected components of raw CELs, vectorizes them into polylines. If in enable_color_masking mode is then connected components of raw color CELs are also computed, and then vectorize them into polylines. The function stp_reference of type Func_stp_top_down_ref is called to learn bonding area and to create the single camera component of the reference including bonding pads windows, masks zones and stable features for registration. Then all transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is bonding_analyze.
chip_area	nop (nop function)	Nothing is done with this window. It is only used within bonding_area window in function stp_reference.

FIG. 7B

Window type	Functions attached	Remarks
balls	<u>balls_ins</u> connect_open (Func_connected_components<CEL>) vectorize (is needed only for debugging and visualization of balls algorithms) Func_angle_vectorizer<CEL>) circles_process (Func_circles_process) trans2ref (Func_trans2ref) forward2balls_analyze (Func_forward)	<p>Produces connected components of raw CELs. Then vectorize them into polylines. This step is not really needed for the algorithm but it provides a nice picture to look at when visualizing the outcome of the algorithm. The function circles_process of type Func_circles_process is called to model the connected components as what are called generalized circles (balls).</p> <p>Then all transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is balls_analyze.</p>
cavity	<u>alarms_lrn</u> watchdog_func (Func_watchdog) trans2ref (Func_trans2ref) forward2ref_if_in_camera (Func_forward)	Function watchdog checks to see if there are CELs inside the window. Any CEL found inside the window is reported as defect. Then all transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally, if the window is completely within camera boundary then the window is forwarded into destination reference and the function that is attached to the window at its new destination is nop. If the window is not within camera boundary then the window is not forwarded.

FIG. 8

Function name	Functions executed	Remarks
balls_analyze	analyze_circles balls_create_ref balls_display (for display and debug only) merge_defects forward2balls_ref (Func_forward)	<p>Classifies circles based on data coming from three cameras. Creates circles reference. Merges all defects from cameras into main unified data.</p> <p>Finally the window is forwarded into destination unified_balls_reference_defects_queue and the function that is attached to the window at its new destination is strip_balls.</p>
bonding_analyze	analyze-bonding-side forward2bonding_ref (Func_forward)	<p>Megre data from all three cameras into unified reference format.</p> <p>Finally the function is forwarded into destination unified_bonding_reference_defects_queue and the function that is attached to the window at its new destination is strip-bonding.</p>
target_analyze	analyze_target forward2target_ref (Func_forward)	<p>Megre data from all three cameras into unified reference format.</p> <p>Finally the function is forwarded into destination unified_reference_defects_queue and the function that is attached to the window at its new destination is strip_target.</p>

FIG. 9

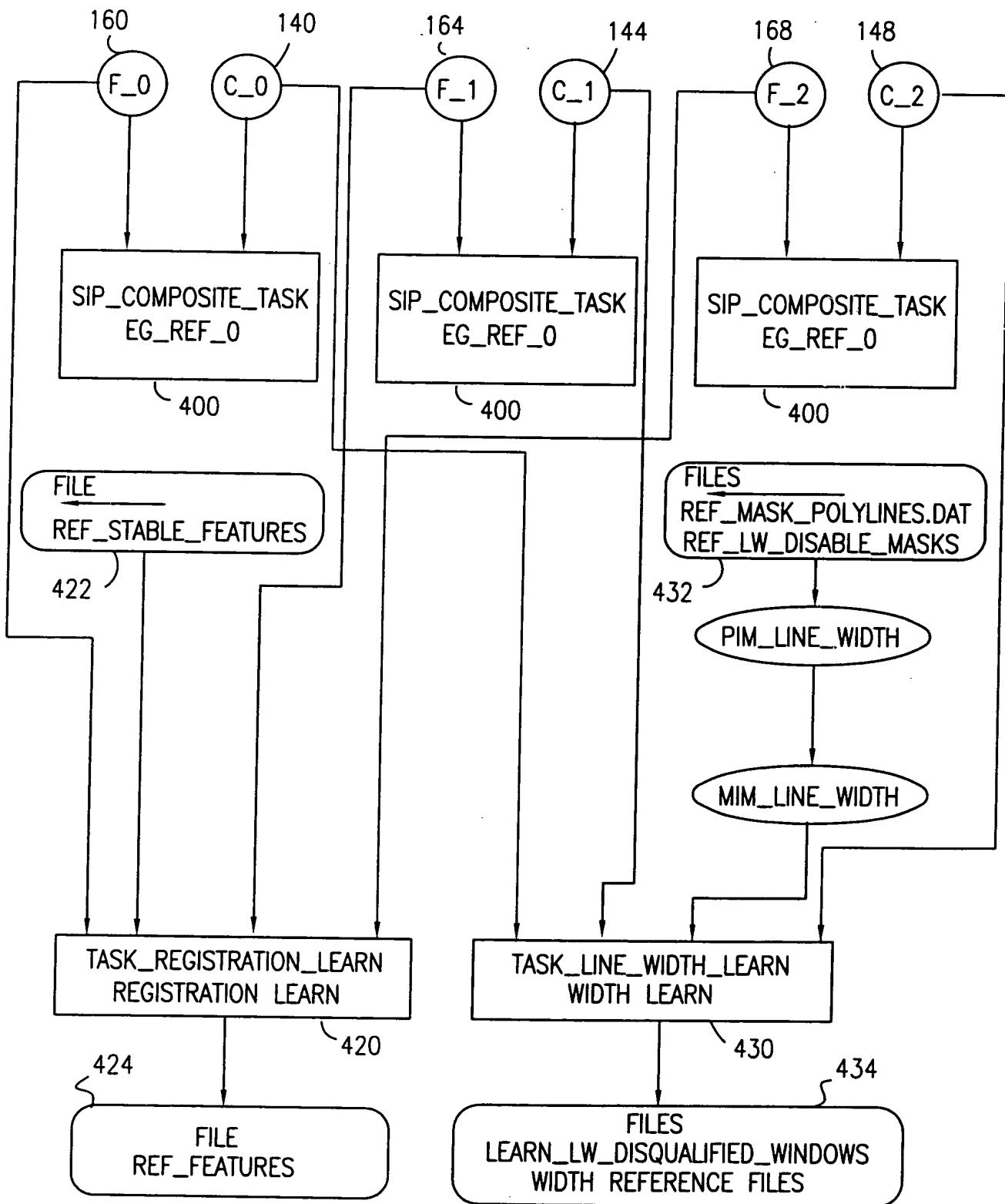


FIG. 10

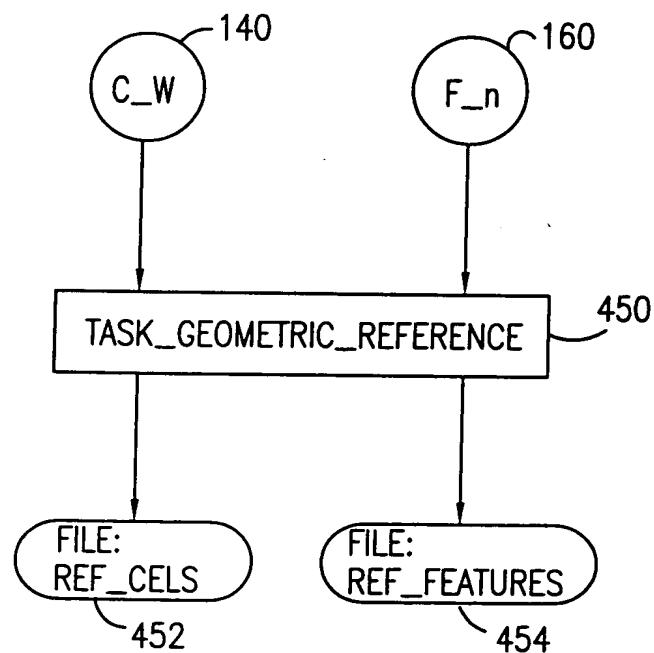


FIG. 11

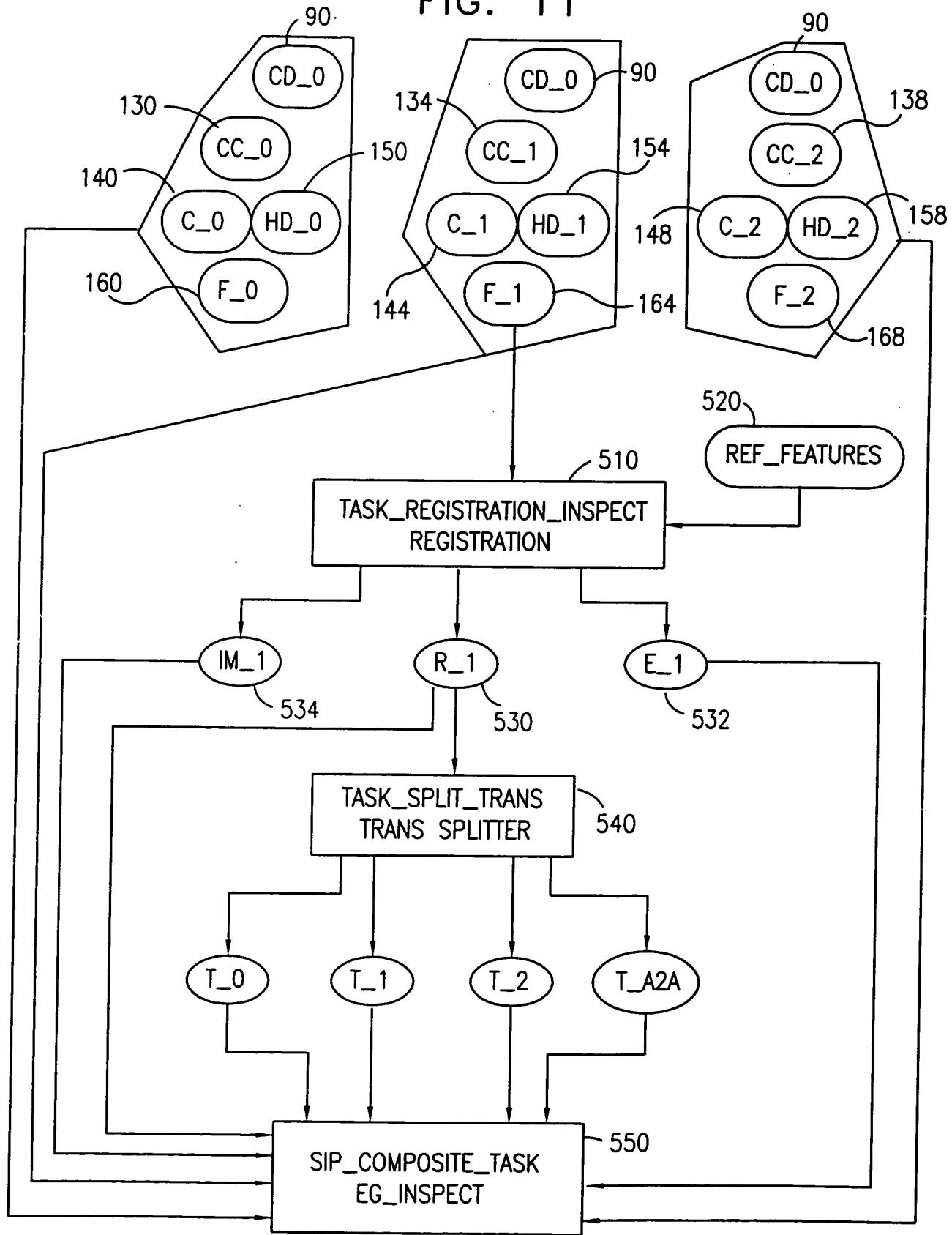


FIG. 12

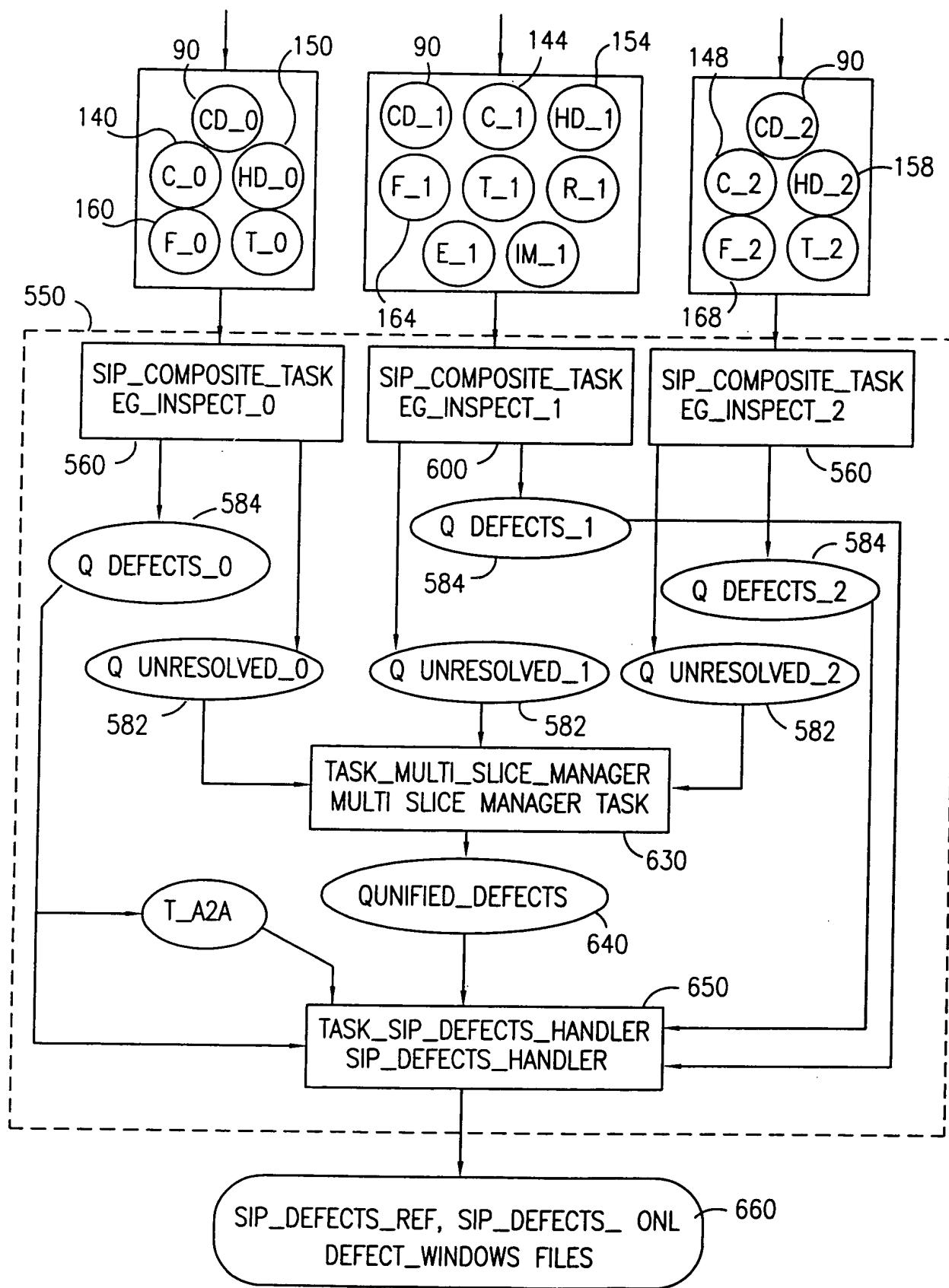


FIG. 13

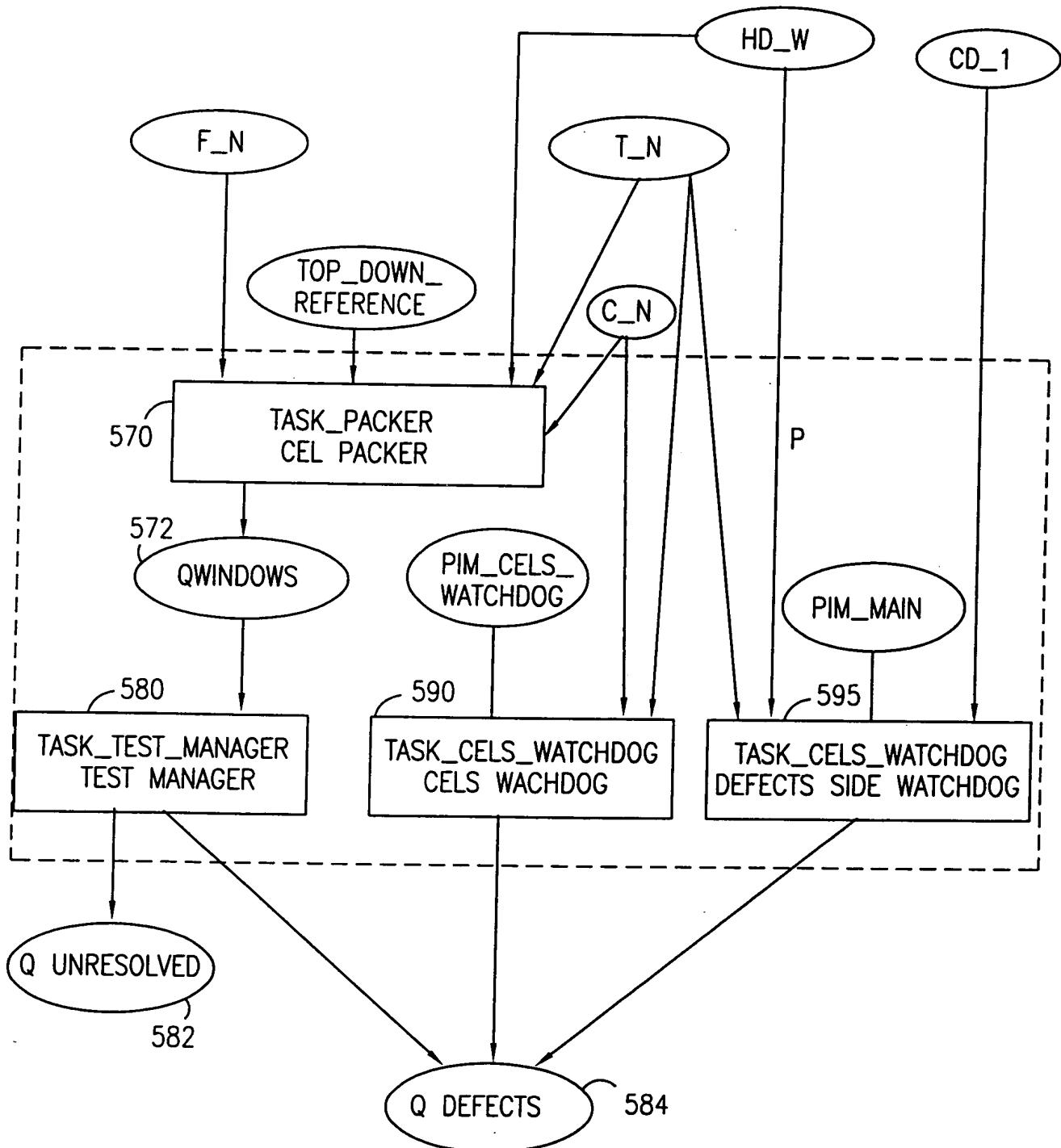


FIG. 14

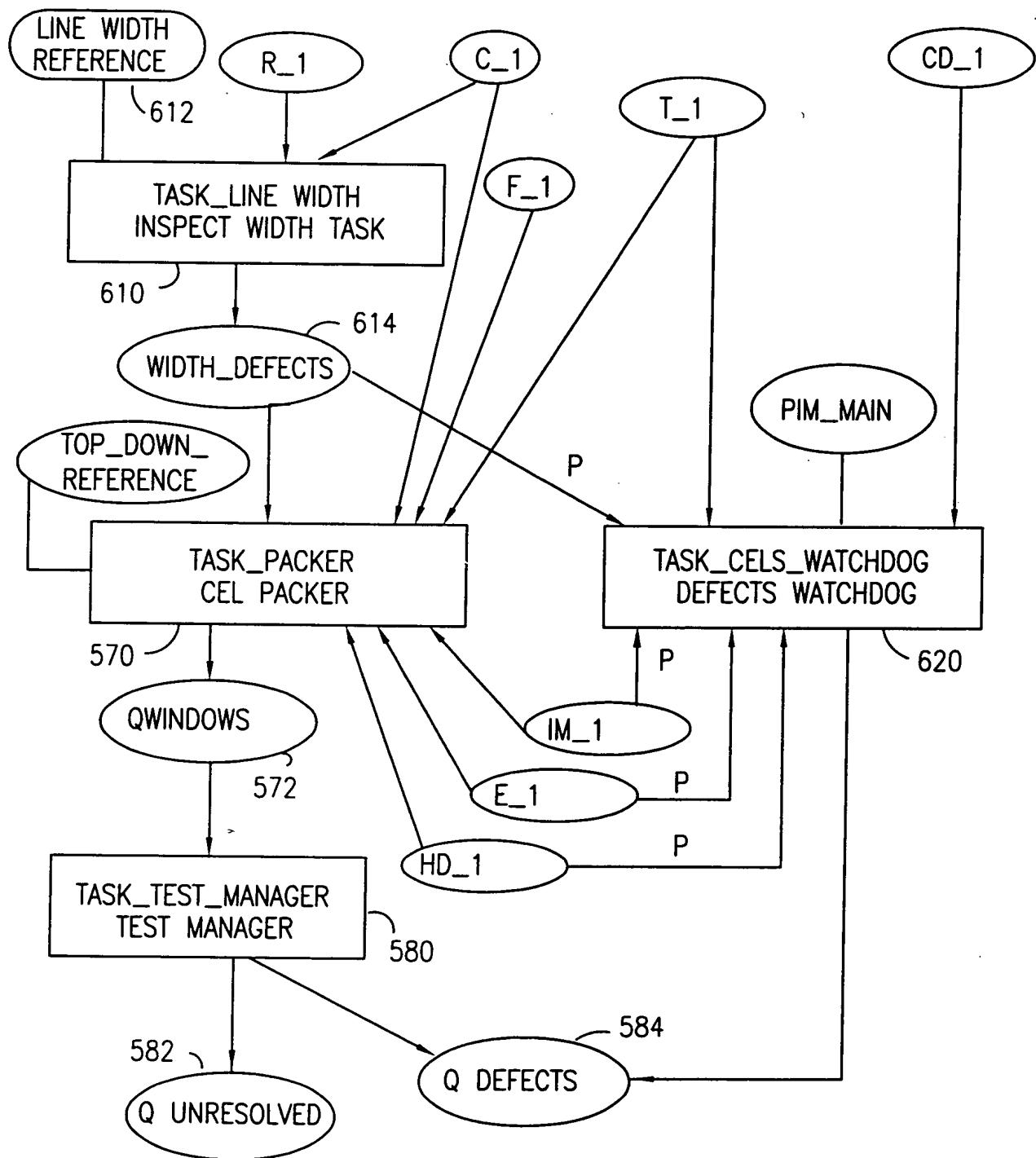


FIG. 15A

Window type	Functions attached	Remarks
target	<p><u>poly compare target:</u></p> <p><code>polylines_comp_ooo_target (func_polylines_comp_ooo)</code></p> <p><code>defects_handler (Func_defects_handler)</code></p> <p><code>defectsfilter (Func_defects_filter)</code></p> <p><code>forward2defects (Func_forward)</code></p>	<p>Function <code>polylines_comp_ooo_target</code> of type <code>func_polylines_comp_ooo</code> do a CEL2VEC comparison between reference polygons and online CELs based on application target tolerances.</p> <p>Function <code>defects_handler</code> of type <code>Func_defects_handler</code> is a post processing function that decides which of the defects reported by the various defect detectors (nick, protrusion, width defect CEL2VEC, excess/missing, etc.) are real Sip_defects. Defects found by a detector are rechecked according to zone specific application criteria.</p> <p>Function <code>defectsfilter</code> is of type <code>Func_defects_filter</code> is called to filter out any defects. It uses <code>PIM_main</code> and filters out all defects found in one or more regions of the following: unstable, mask_region, power_line_default.</p> <p>Finally, if any Sip_defects are inside the window, then the window is forwarded into destination defects and the function that is attached to the window at its new dimension is <code>nop</code> (nothing to do). If no defects are found, then the window is not forwarded at all.</p>
pad_ref	<p><u>poly compare pads</u></p> <p><code>polylines_comp_ooo_target (func_polylines_comp_ooo)</code></p> <p><code>defects_handler (Func_defects_handler)</code></p> <p><code>defectsfilter(Func_defects_filter)</code></p> <p><code>forward2defects(Func_forward)</code></p>	<p>This function connected to this window type is similar to the function connected to the target window above. The only difference is that parameter applications are used for bonding pads to control the behavior of the CEL2VEC function.</p>

FIG. 15B

Window type	Functions attached	Remarks
balls	<u>balls_ins</u> connect_open (Func_connected_components<CEL>) vectorize (is needed only for debugging and visualization of balls algorithms) Func_angle_vectorizer<CEL> circles_process (Func_circles_process) trans2ref (Func_trans2ref) forward2balls_analyze (Func_forward)	<p>Produces connected components of raw CELs. Then they are vectorized into polylines. This step is not really needed for the algorithm but it provides a nice picture to look at when visualizing the outcome of the algorithm. The function circles_process of type Func_circles_process is called to model the connected components and are called generalized circles (balls). Then all the transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is balls_analyze.</p>
cavity	<u>watchdog</u> watchdog_func (Func_watchdog) defectsfilter (Func_defects_filter) forward2defects (Func_forward)	<p>Function watchdog checks to see if there are CELs inside the window. Any CEL which is found inside the window is reported as a defect.</p> <p>Function defectsfilter is of type Func_defects_filter is called to filter out any defects. It uses PIM_main and filters out all defects found in one or more regions of the following: unstable, mask_region, power_line_default.</p> <p>Finally, if any Sip_defects are found inside the window, then the window is forwarded into destination defects and the function that is attached to the window at its new destination is nop (nothing to do). If no defects are found, then the window is not forwarded at all.</p>

FIG. 15C

Window type	Functions attached	Remarks
reject	<p>poly compare and reject</p> <p>polylines_comp_and_reject (func_polyline_comp_aaa)</p> <p>Forward2defects (Func_forward)</p>	<p>Function polylines_comp_and_reject does a CEL2VEC comparison between reference polygons and online CELs based on application target tolerances. This function has a very small limit to the number of excess CELs or missing envelopes permitted. If there is even a small change between reference target and online target overflow type defects are obtained which are interpreted by the application as an indication that this frame should be rejected.</p> <p>Finally, if any Sip_defects is found inside the window, then the window is forwarded into destination defects and the function that is attached to the window at its new destination is nop (nothing to do). If no defects are found, then the window is not forwarded at all.</p>
disqualified_lw_win	nop	Do nothing

FIG. 16

Function name	Functions executed	Remarks
balls_analyze	analyze_circles (Func_circles_analyze) balls_compare2ref (Func_compare2ref) balls_display (for display and debug only. Func_display_balls_info) Merge_defects (Func_merge_defects) forward2defects (Func_forward)	<p>Classify circles based on data coming from three cameras. Compare unified circles to reference. Merge all defects from cameras into main unified data.</p> <p>Finally, if any Sip defects are found inside the window, then the function is forwarded into destination defects and the function that is attached to the window as its new destination is nop (nothing to do). If no defects are found, then the window is not forwarded at all.</p>

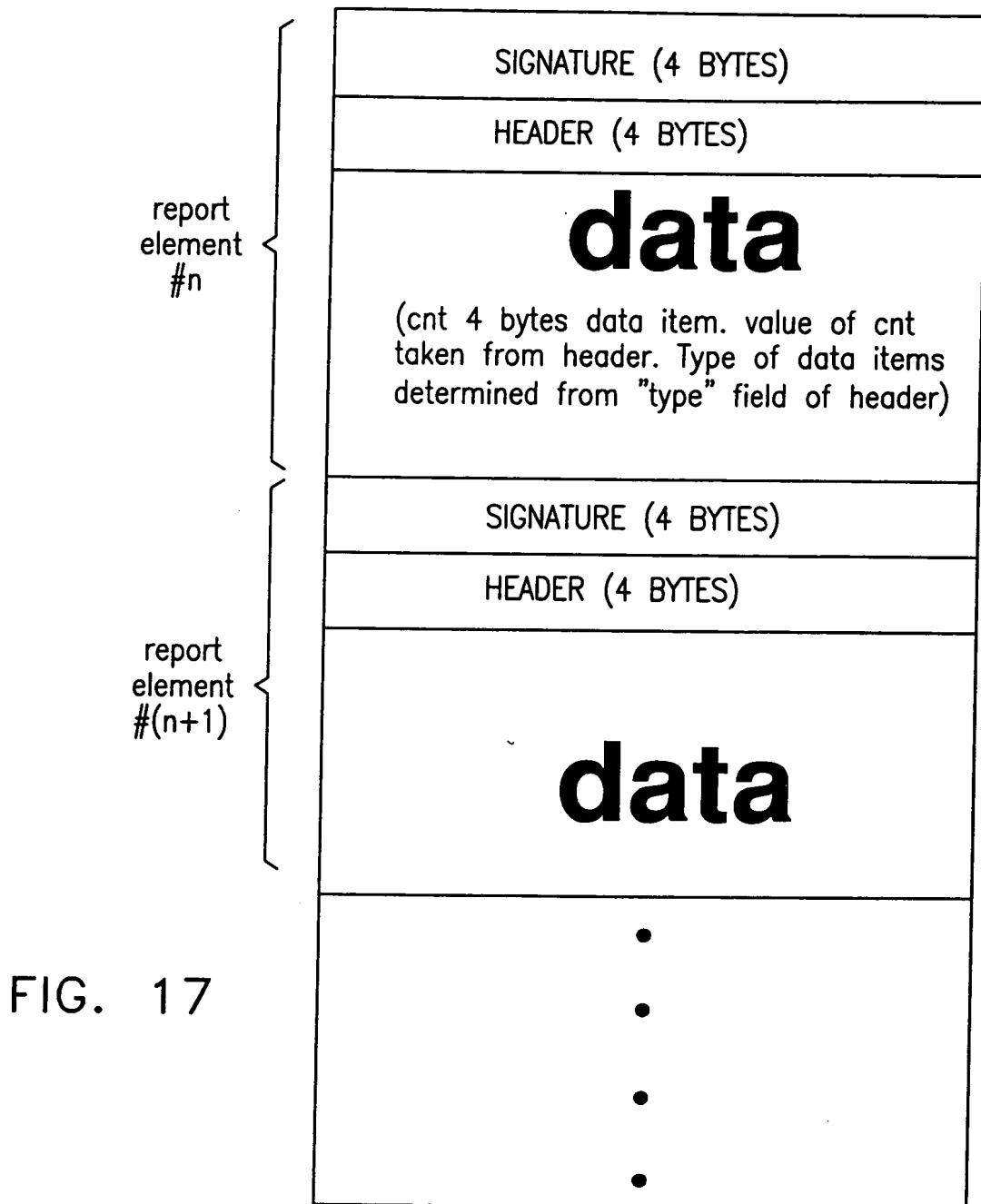


FIG. 17

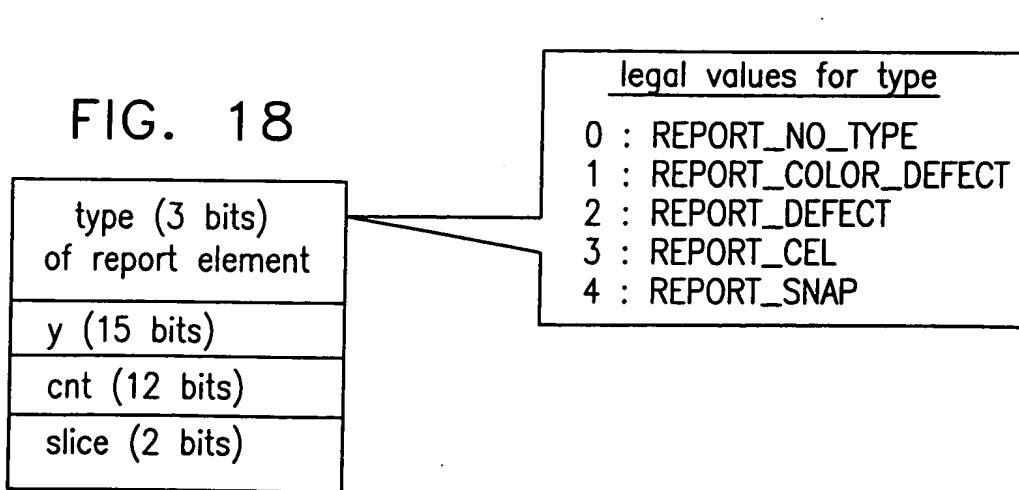


FIG. 19

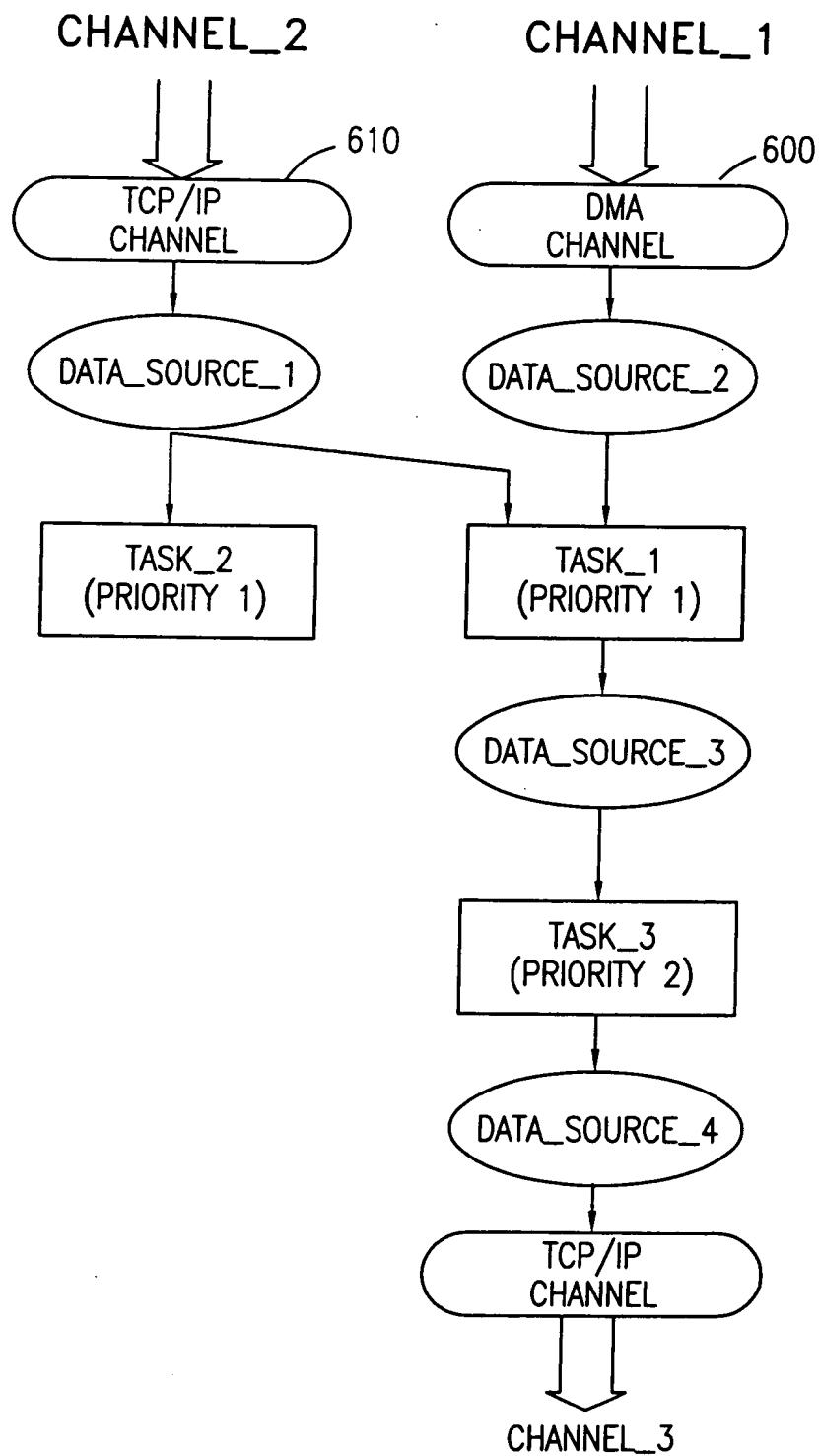
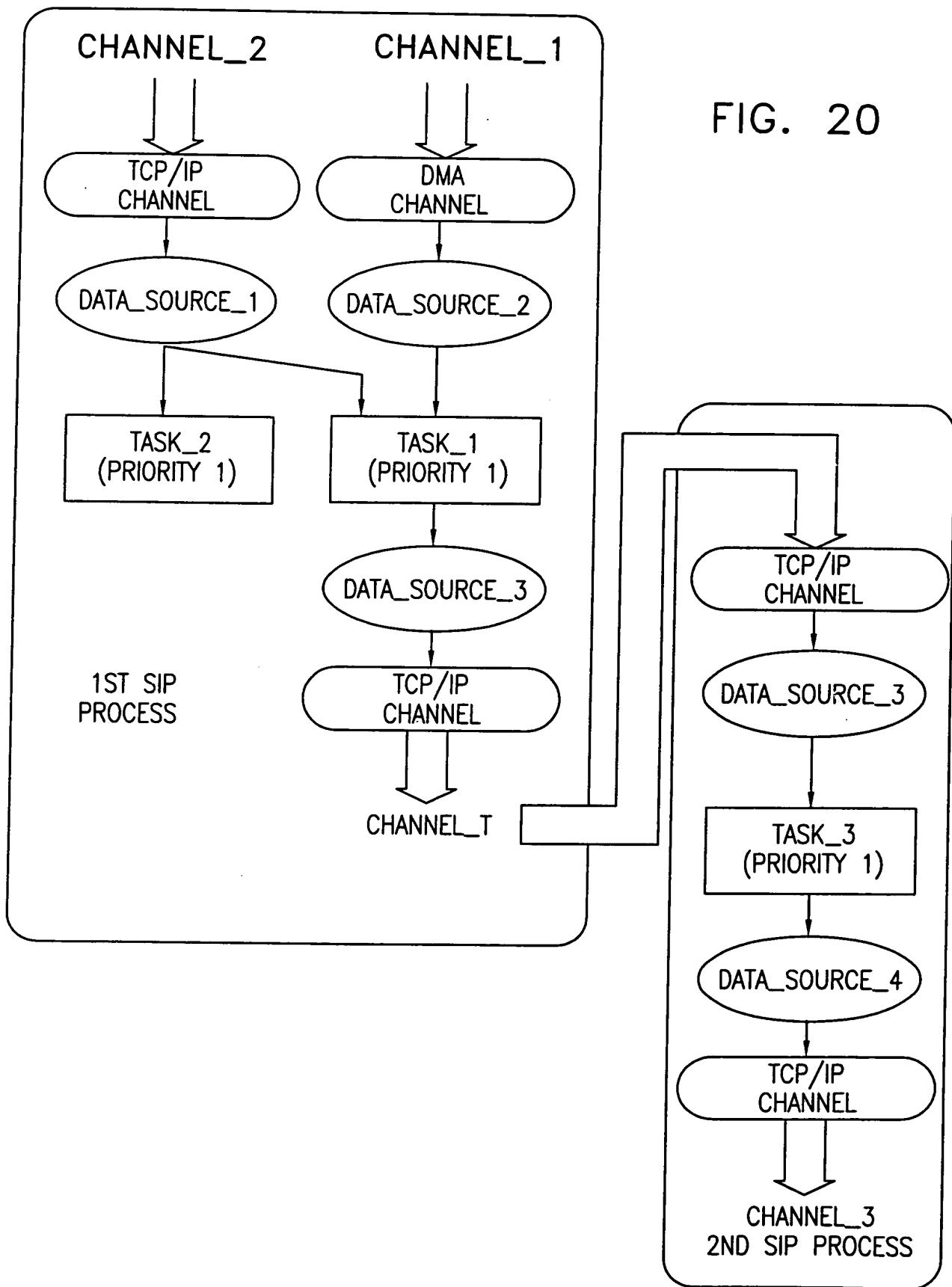


FIG. 20



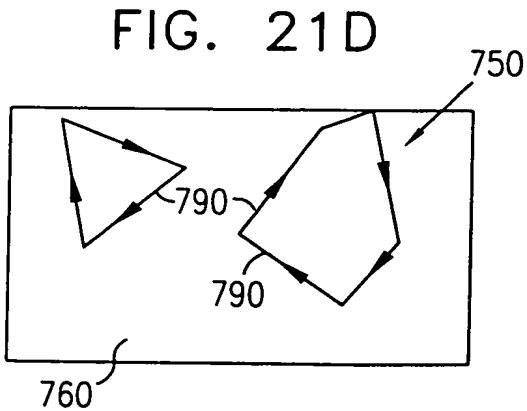
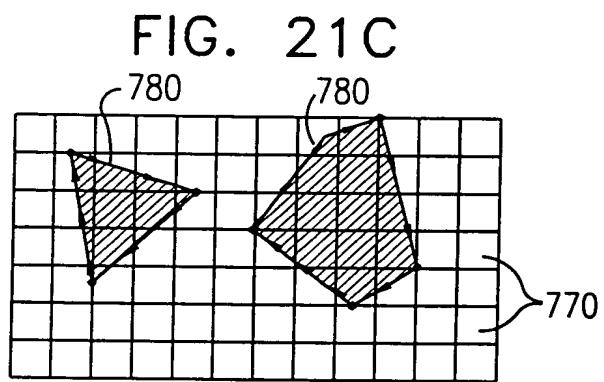
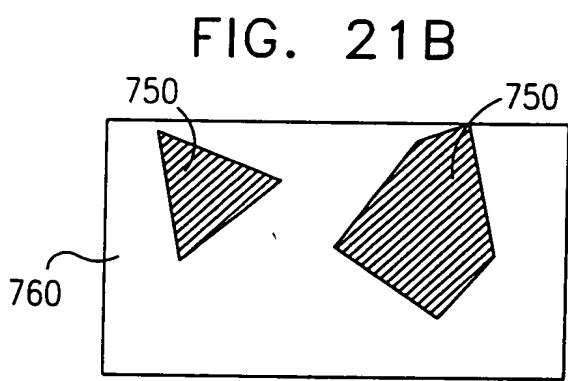
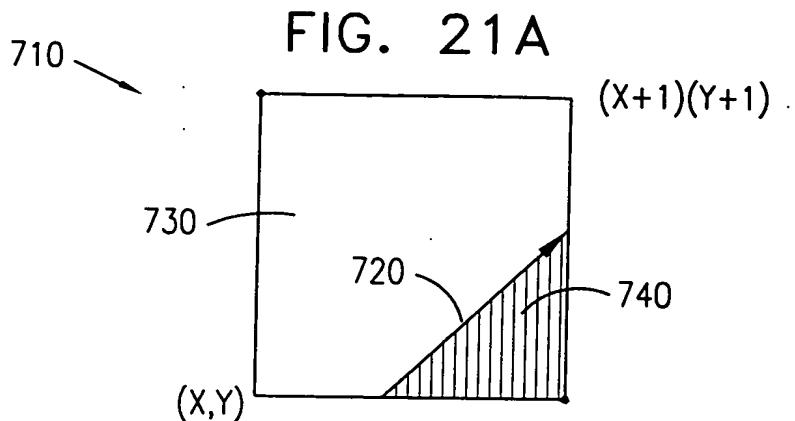


FIG. 22A

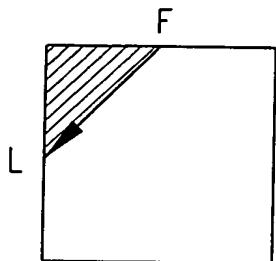


FIG. 22B

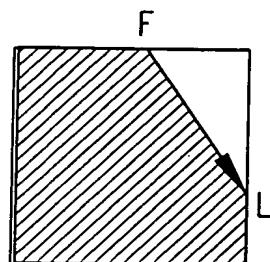


FIG. 22C

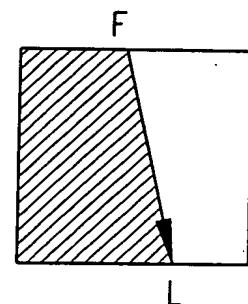


FIG. 22D

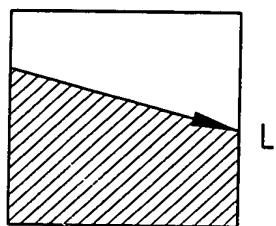


FIG. 22E

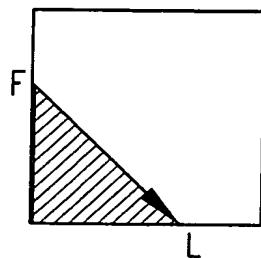


FIG. 22F

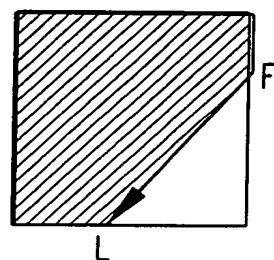


FIG. 22G

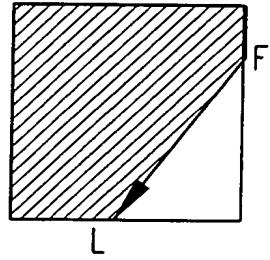


FIG. 22H

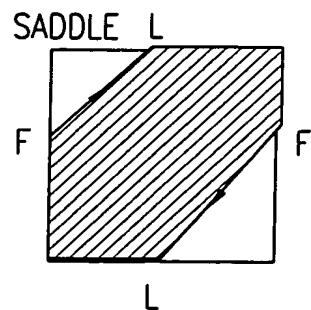


FIG. 23A

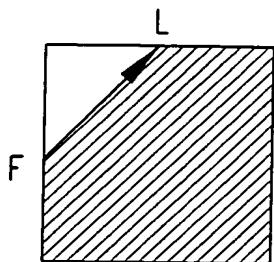


FIG. 23B

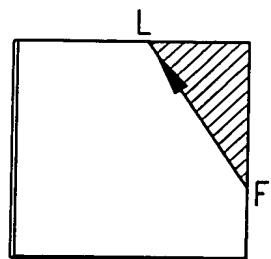


FIG. 23C

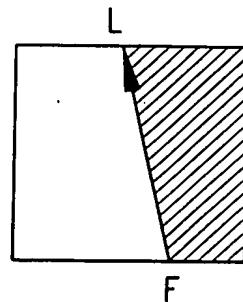


FIG. 23D

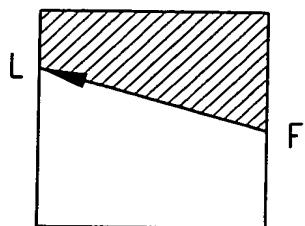


FIG. 23E

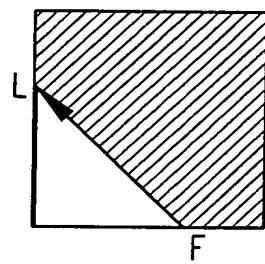


FIG. 23F

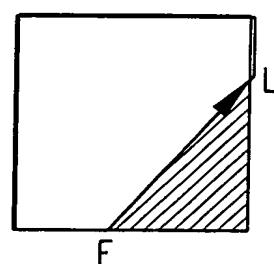


FIG. 23G

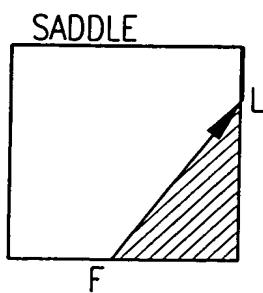


FIG. 23H

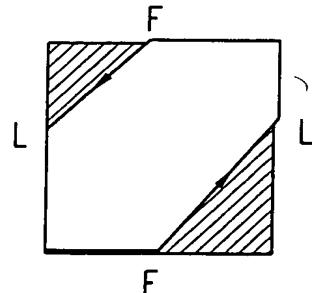


FIG. 23I

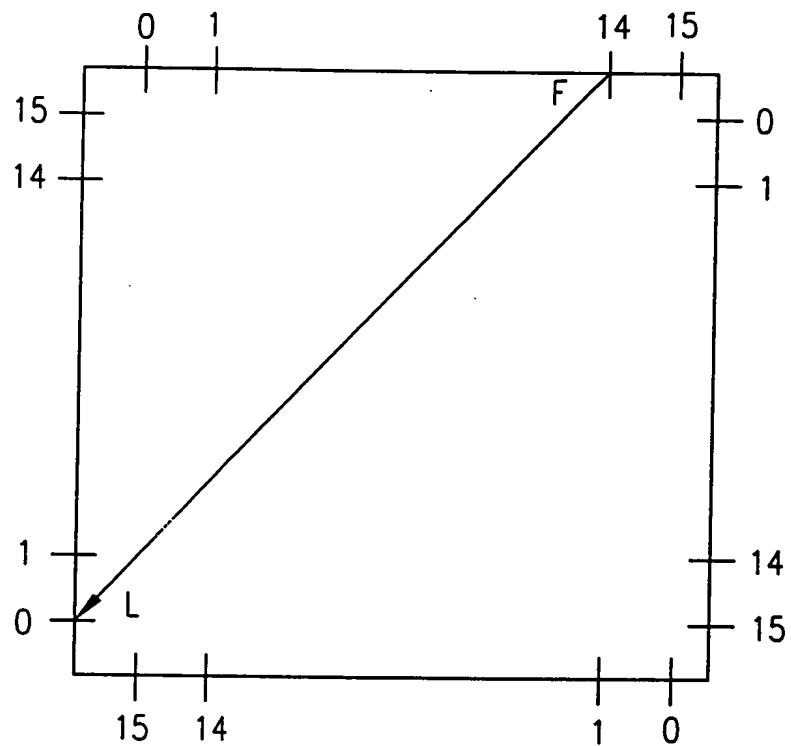


FIG. 24

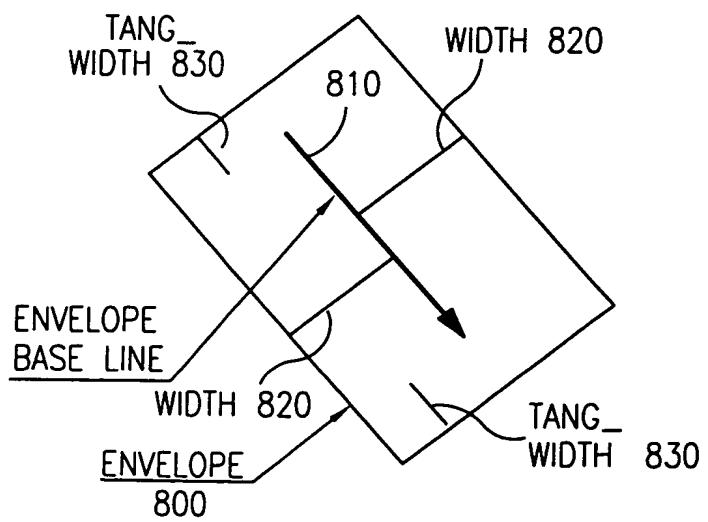


FIG. 25

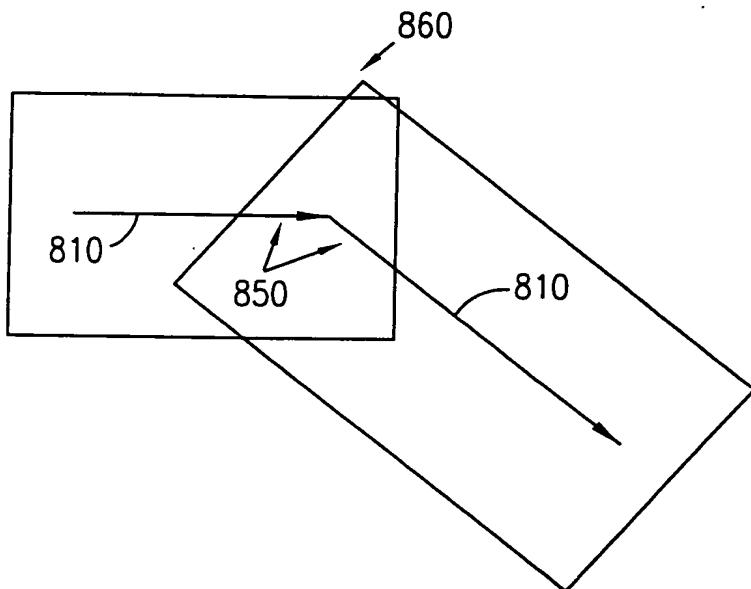


FIG. 26

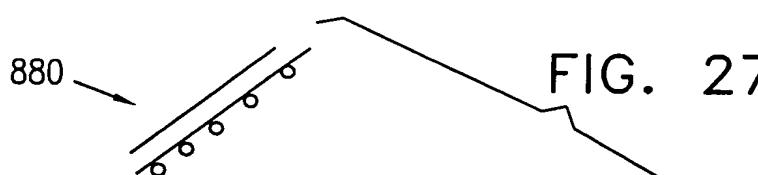
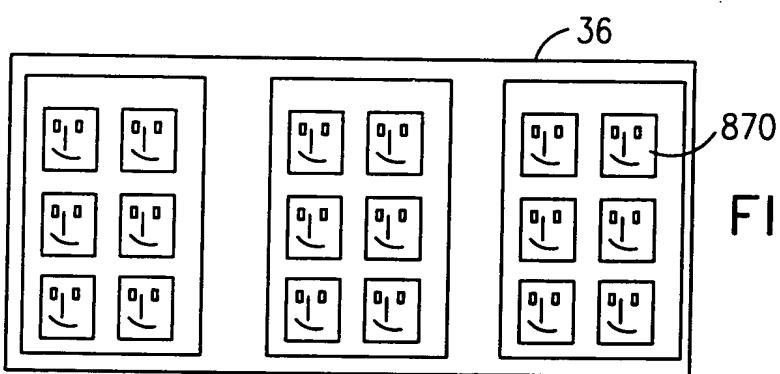


FIG. 27

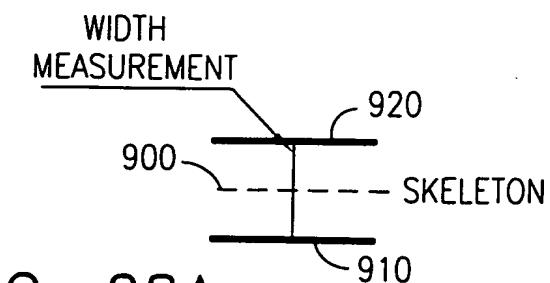


FIG. 28A

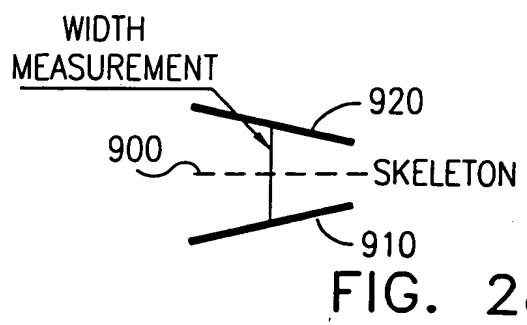
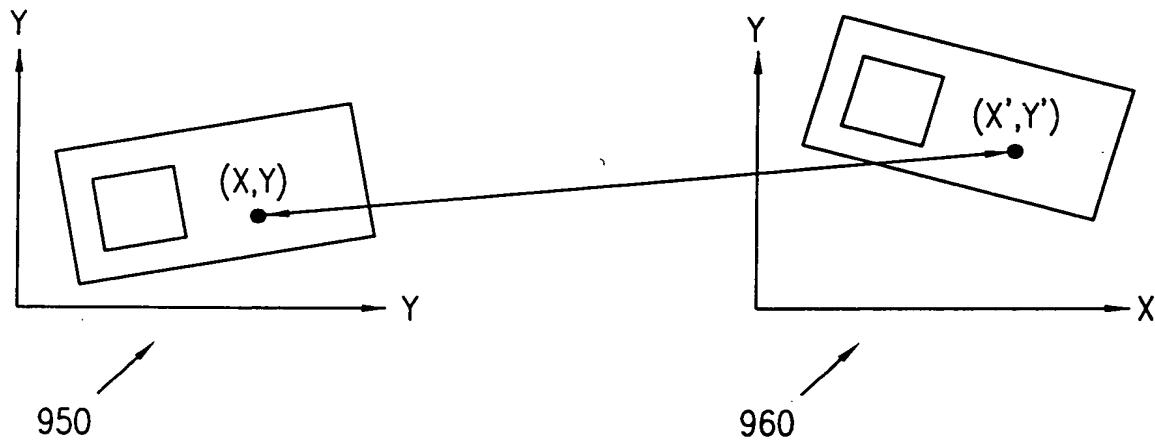


FIG. 28B

FIG. 29



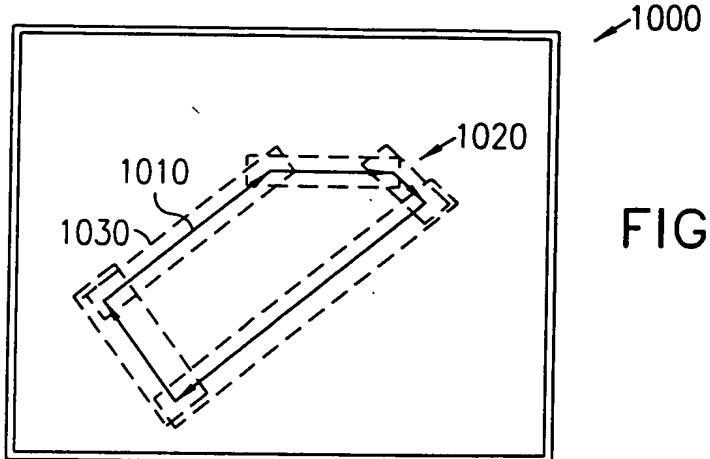


FIG. 30A

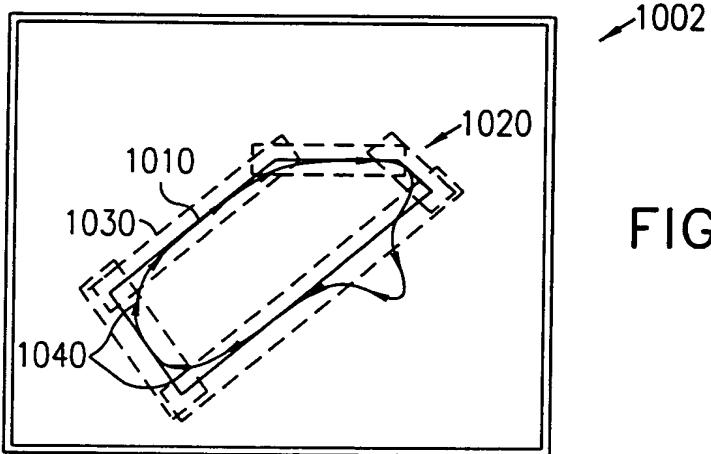


FIG. 30B

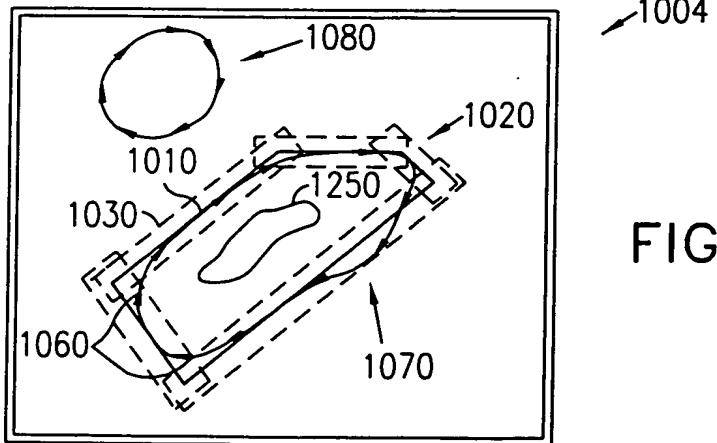


FIG. 30C

FIG. 31

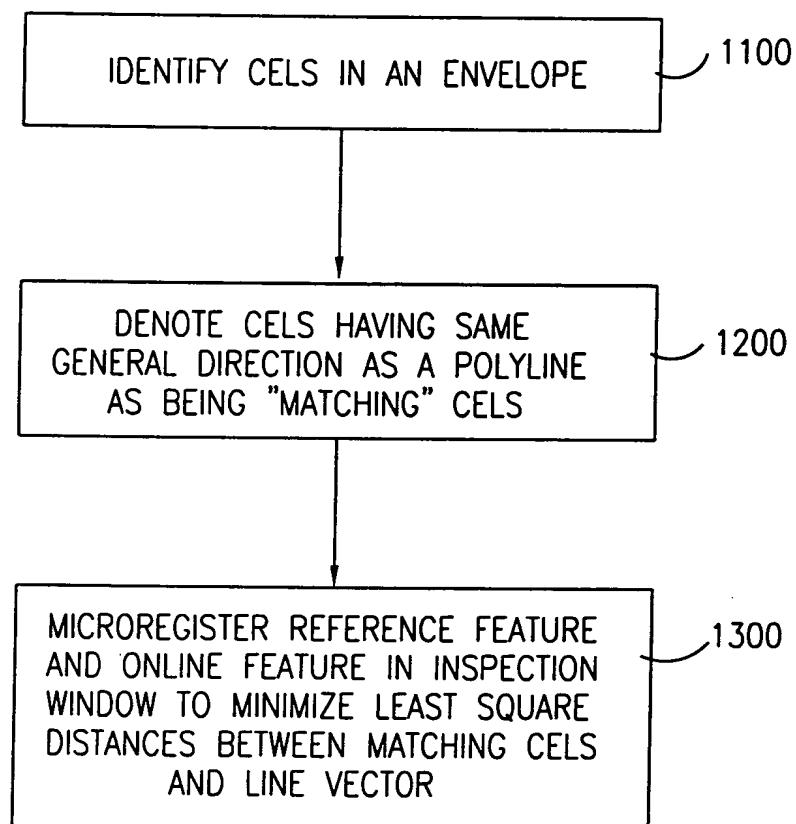


FIG. 32A

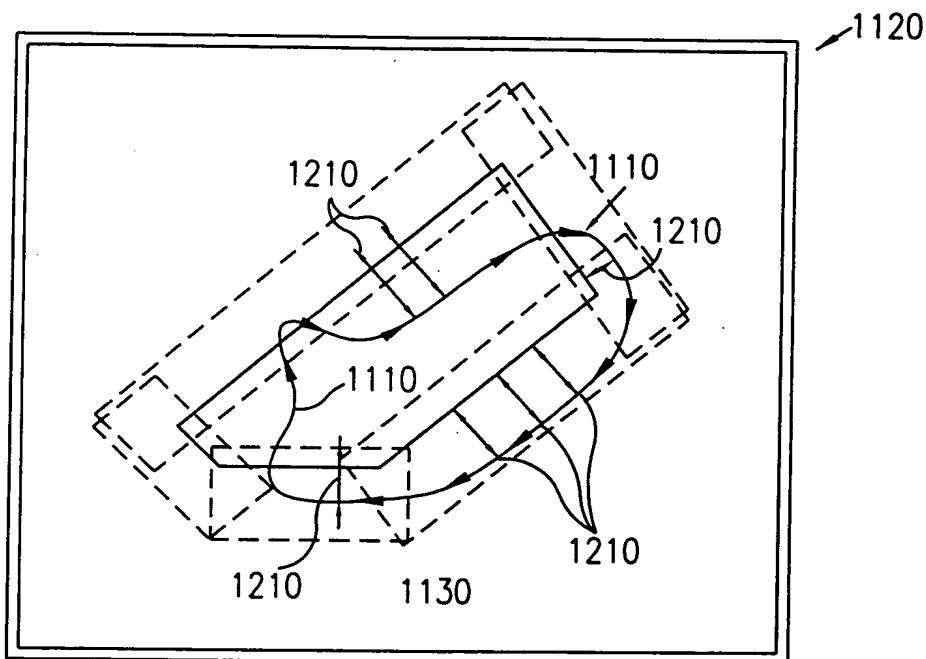


FIG. 32B

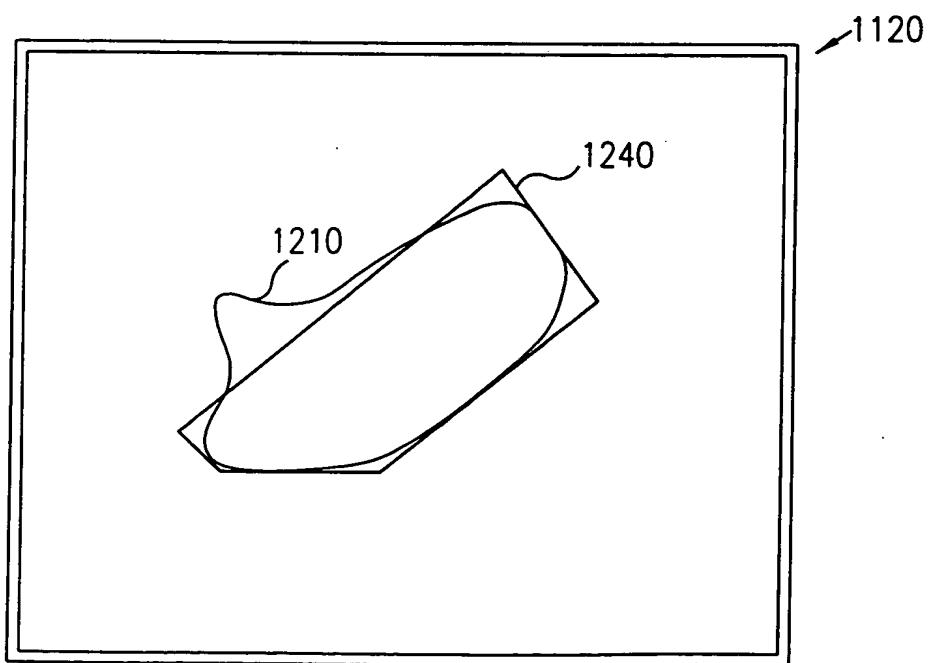


FIG. 33

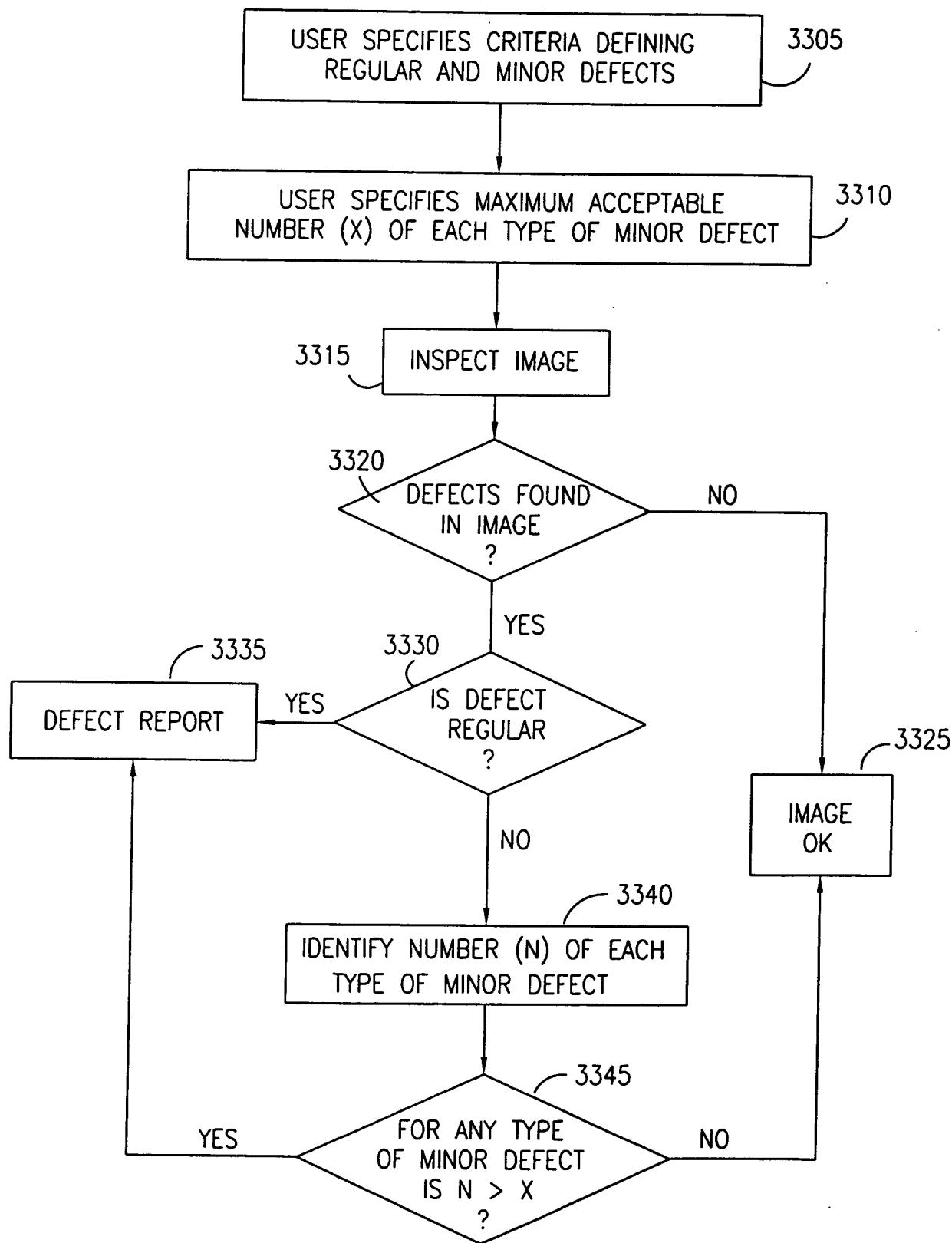


FIG. 34

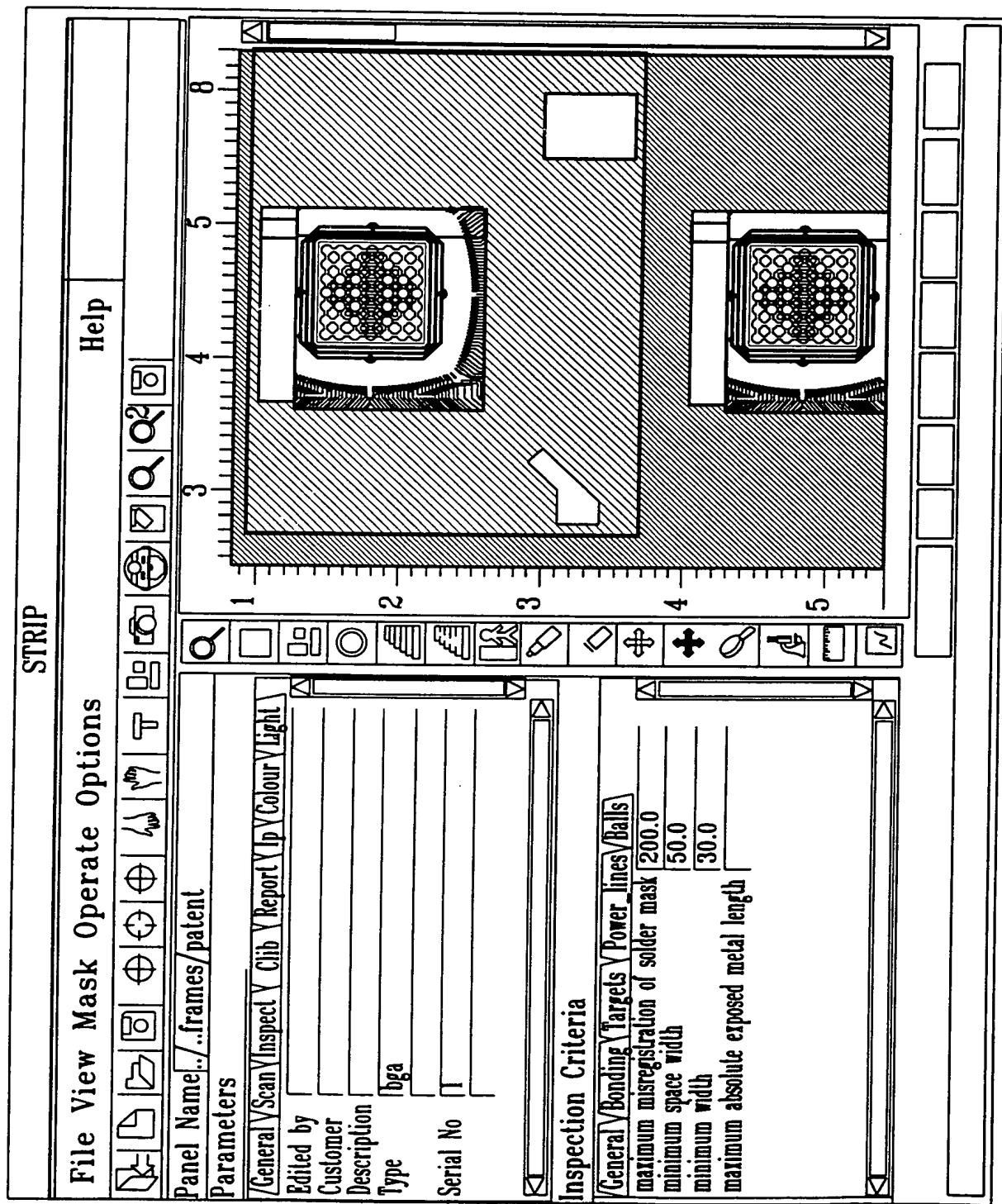


FIG. 35

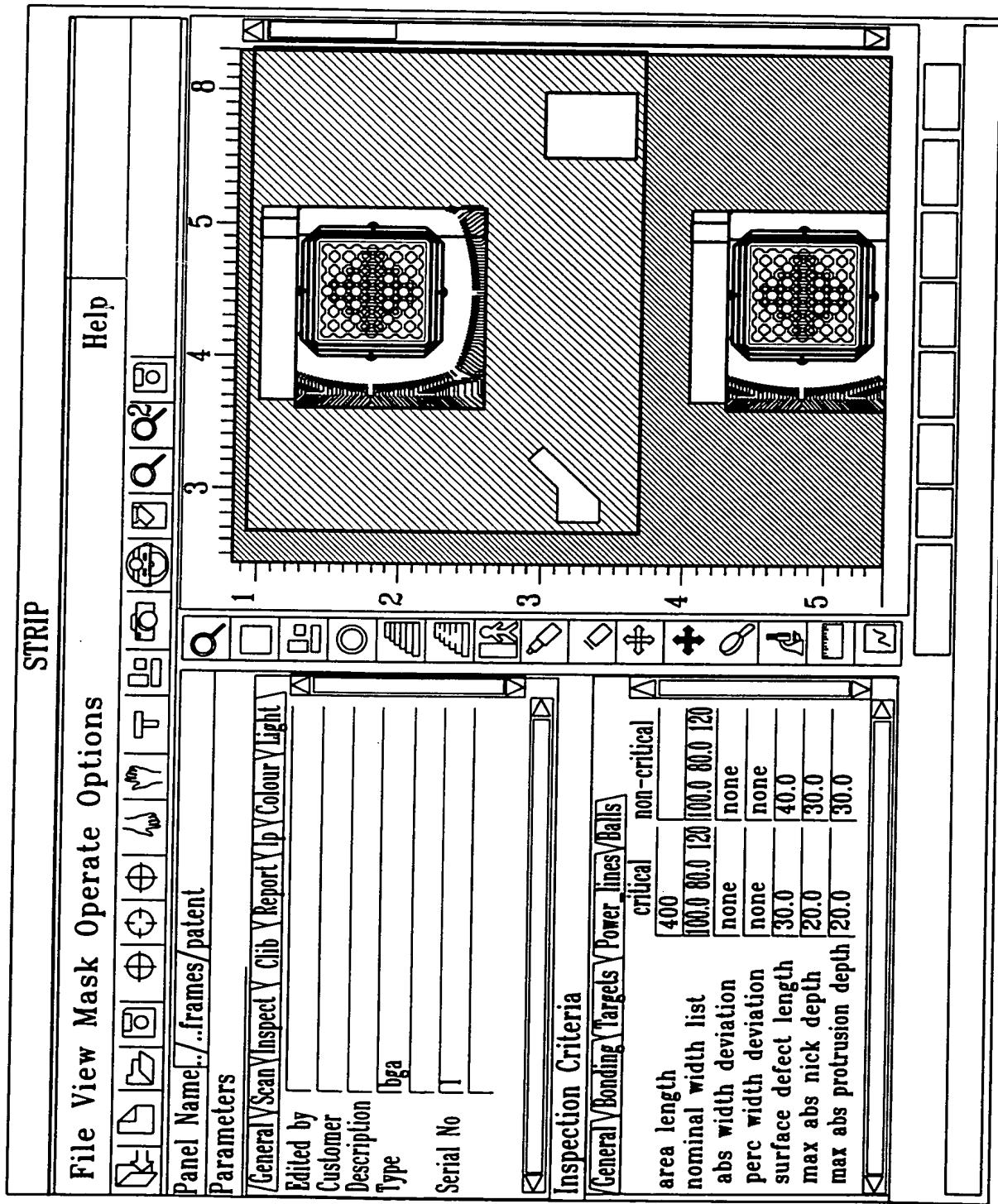


FIG. 36

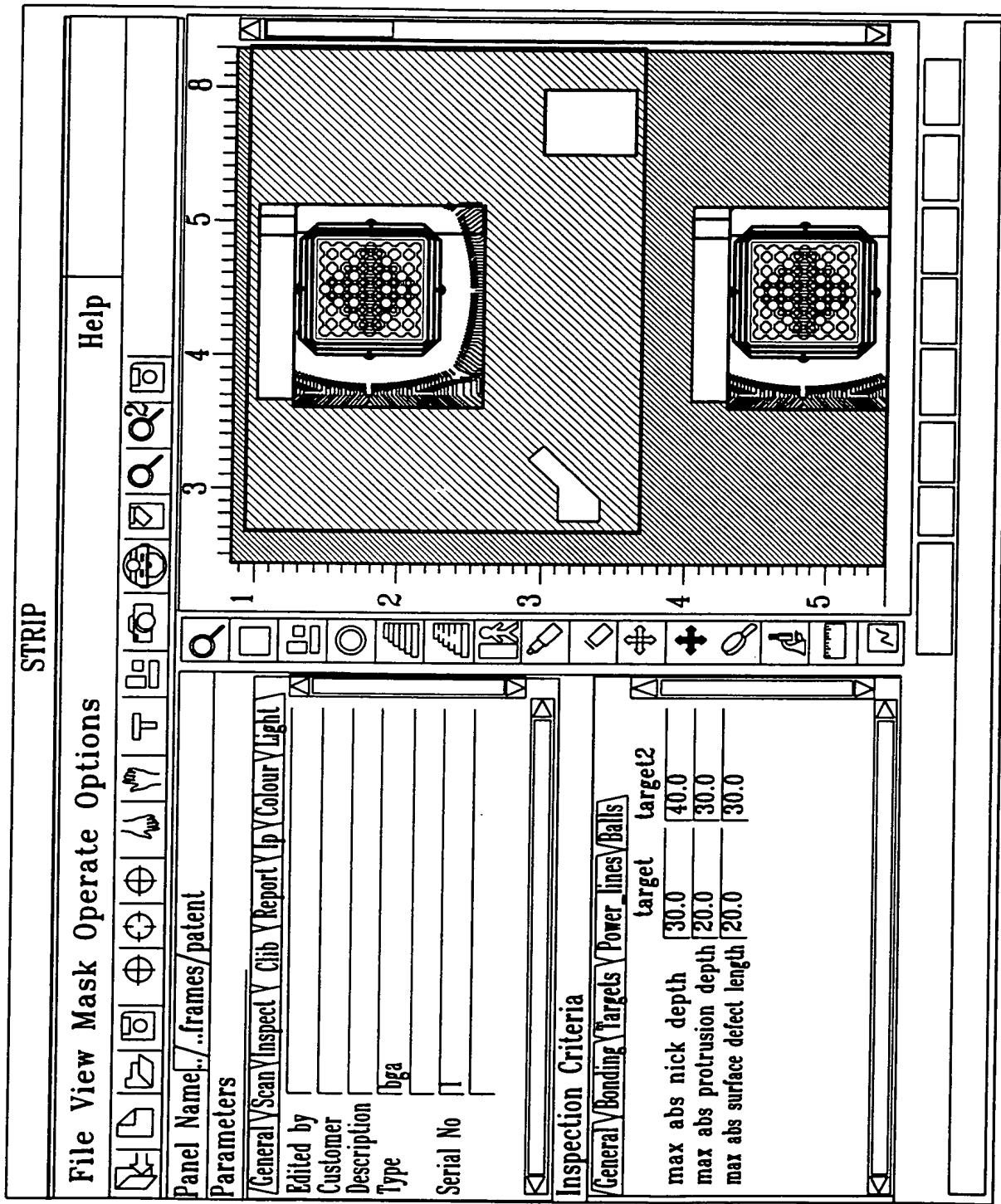


FIG. 37

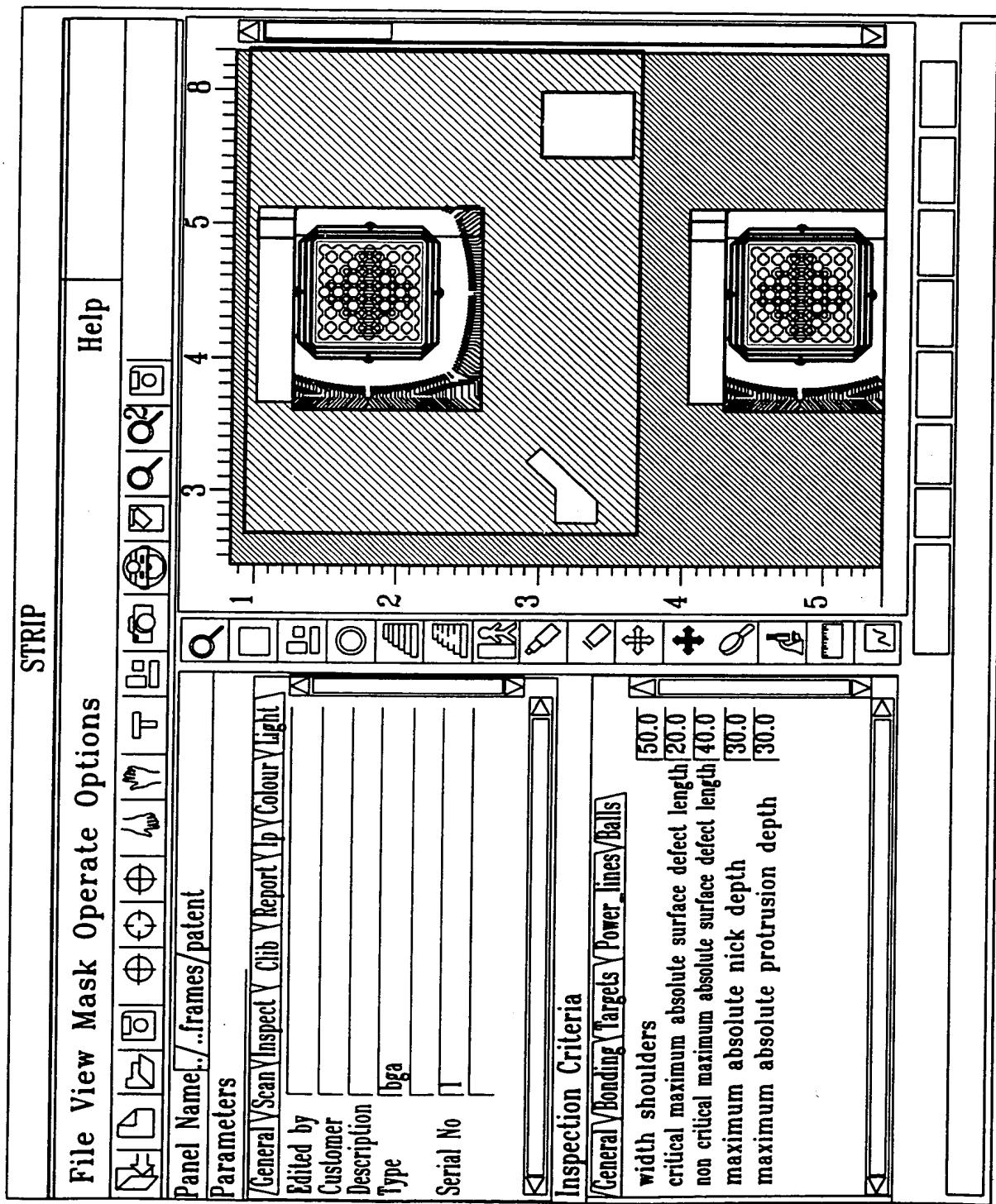


FIG. 38

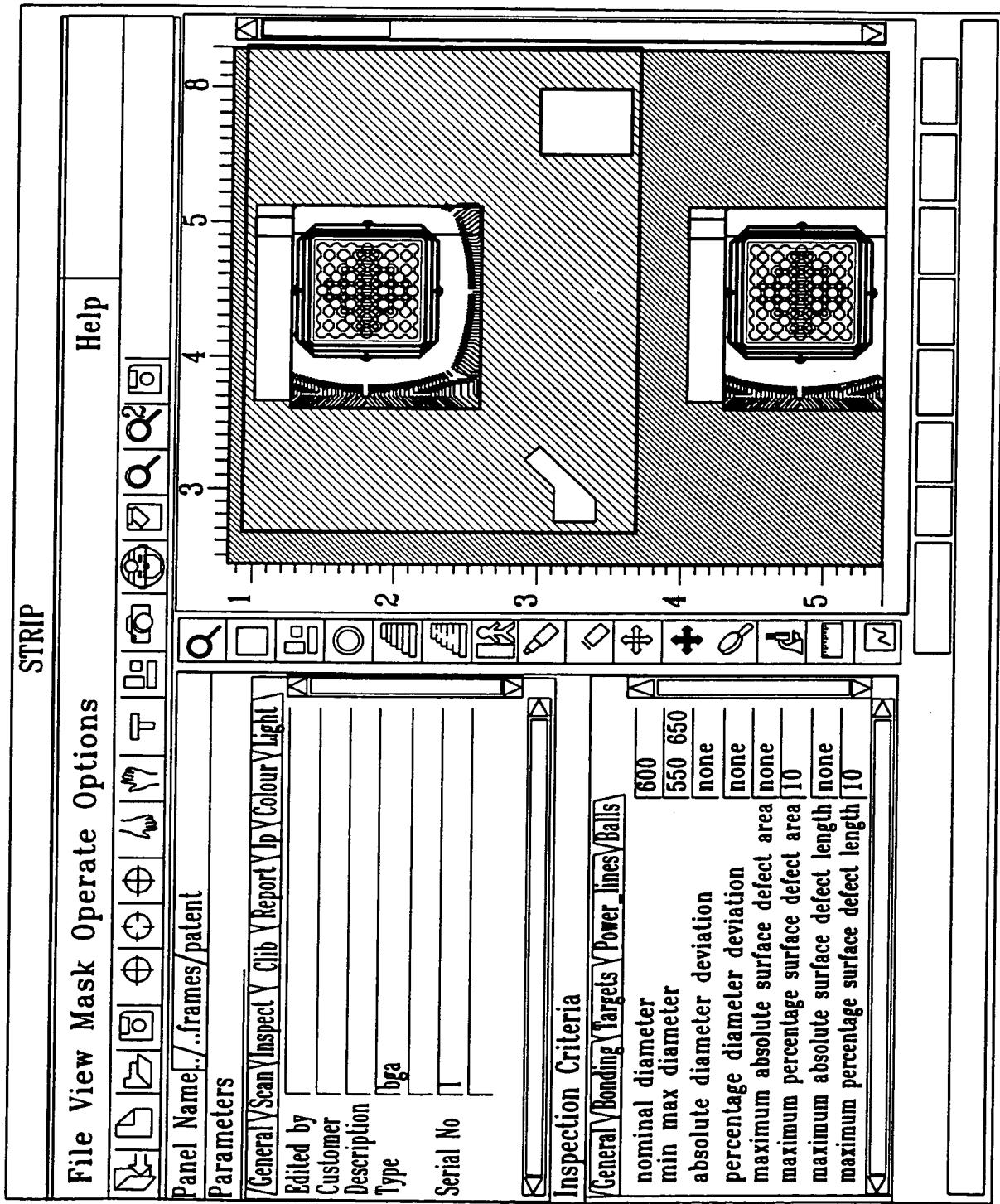


FIG. 39

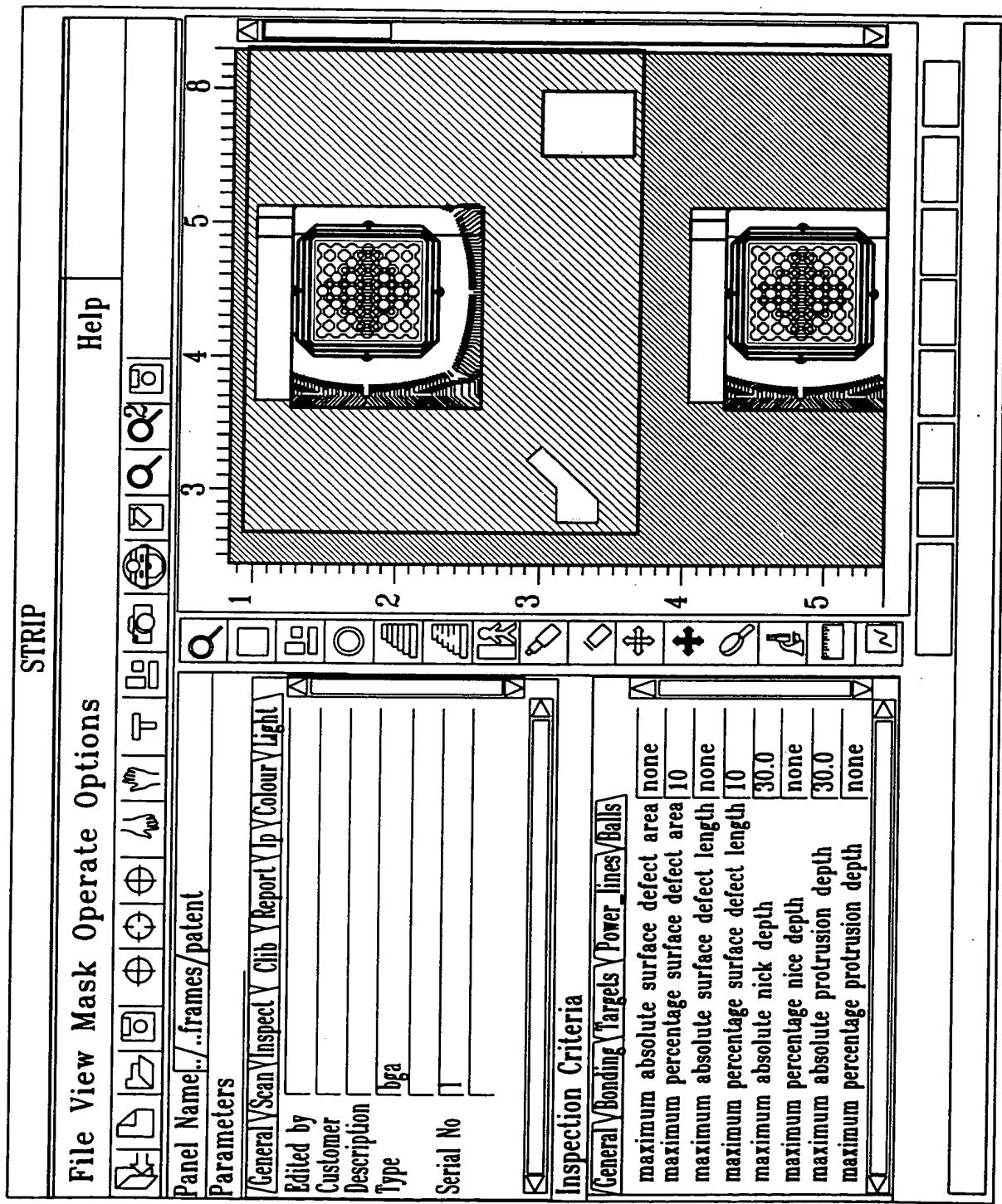


FIG. 40

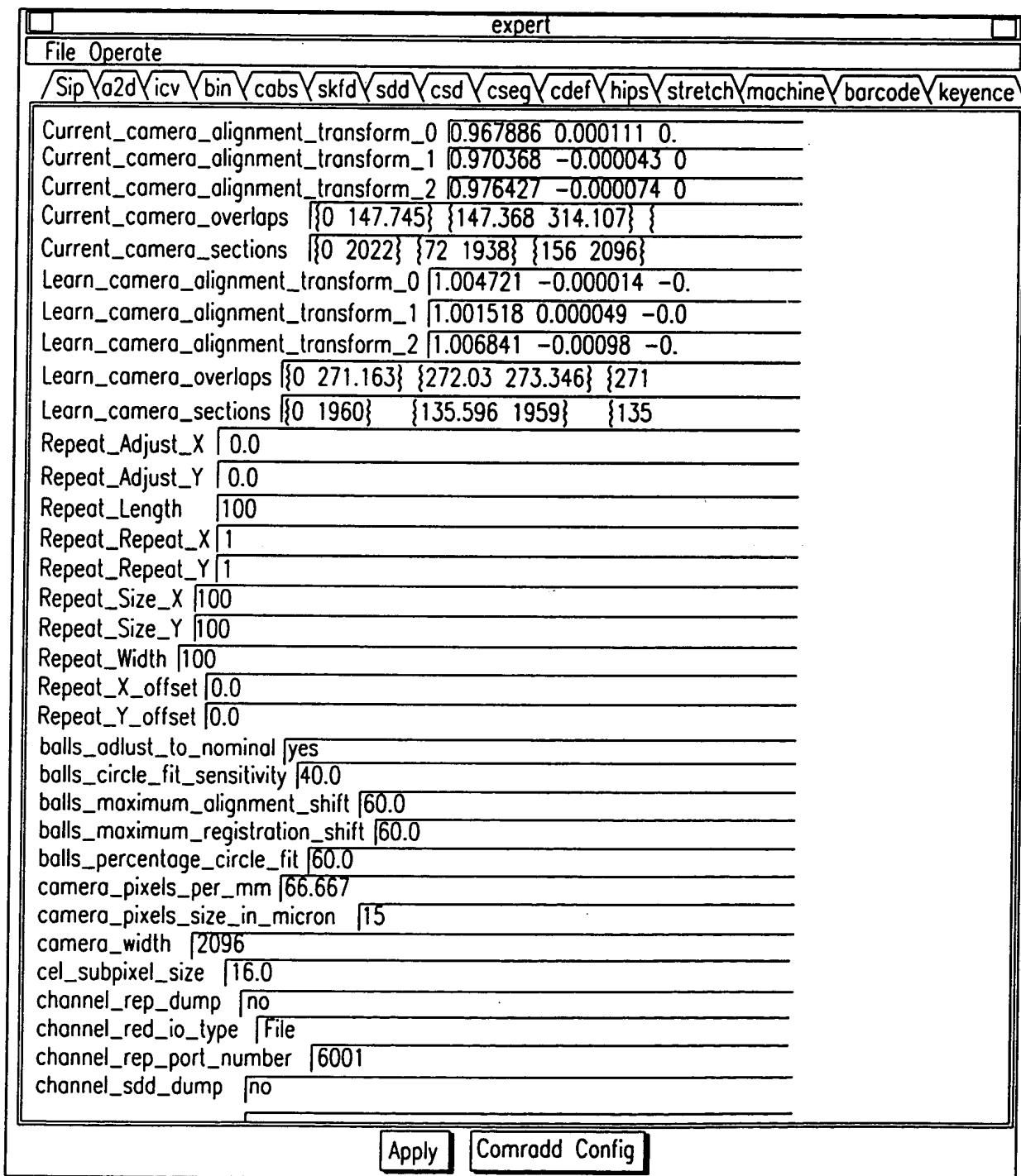


FIG. 41

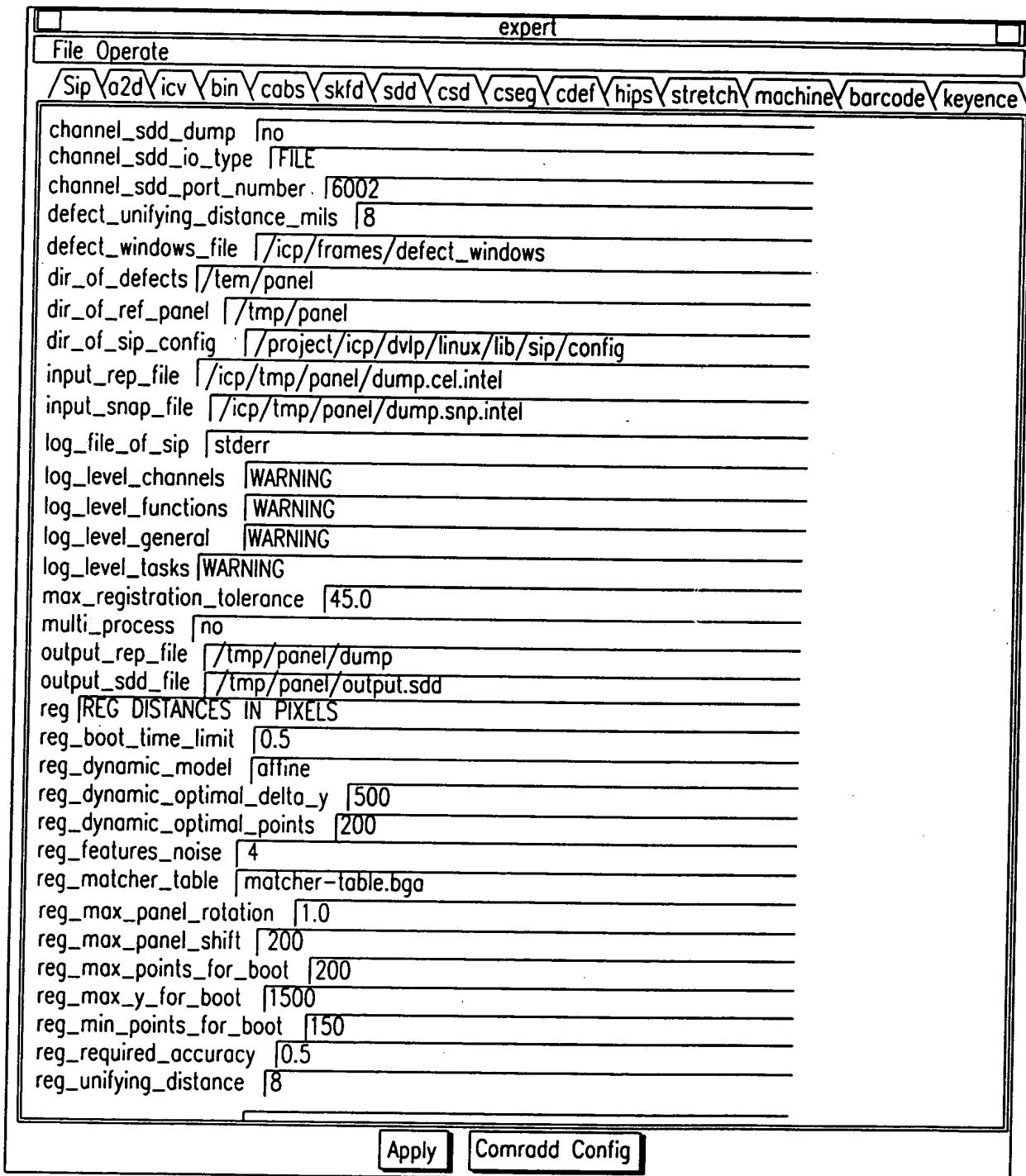


FIG. 42

expert

File Operate	
Sip\	a2d\icv\bin\cabs\skfd\sdd\csd\cseg\cdef\hips\stretch\machine\barcode\keyence
log_level_functions	WARNING
log_level_general	WARNING
log_level_tasks	WARNING
max_registration_tolerance	45.0
multi_process	no
output_rep_file	/tmp/panel/dump
output_sdd_file	/tmp/panel/output.sdd
reg [REG DISTANCES IN PIXELS]	
reg_boot_time_limit	0.5
reg_dynamic_model	affine
reg_dynamic_optimal_delta_y	500
reg_dynamic_optimal_points	200
reg_features_noise	4
reg_matcher_table	matcher-table.bga
reg_max_panel_rotation	1.0
reg_max_panel_shift	200
reg_max_points_for_boot	200
reg_max_y_for_boot	1500
reg_min_points_for_boot	150
reg_required_accuracy	0.5
reg_unifying_distance	8
show_all_em_defects	false
show_all_hw_defects	false
show_all_lw_defects	false
sip_buffer_size	10000000
sip_config_dir	/home/malcolm/lcp/Dvlp/alfi/lib/sip/config
sip_dma_buffer_size	10000000
sip_max_line_in_scan	14664
sip_max_number_of_defect_report	1000
type_enable_color_masking	yes
type_manual_pads_learning	no
type_panel_polarity	1
type_power_lines_area_threshold	500.0

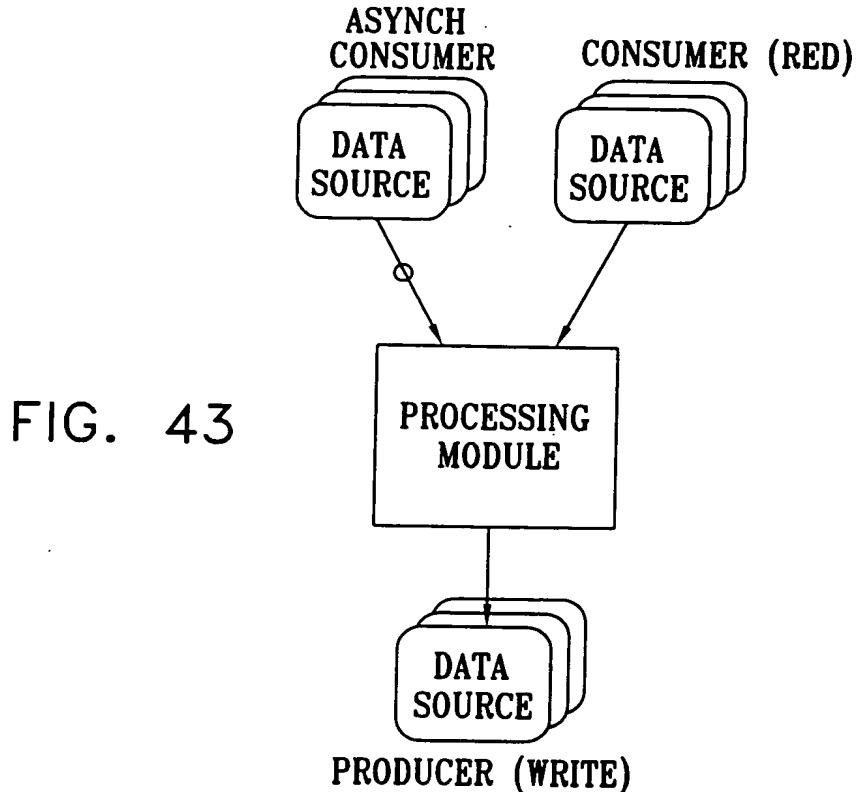
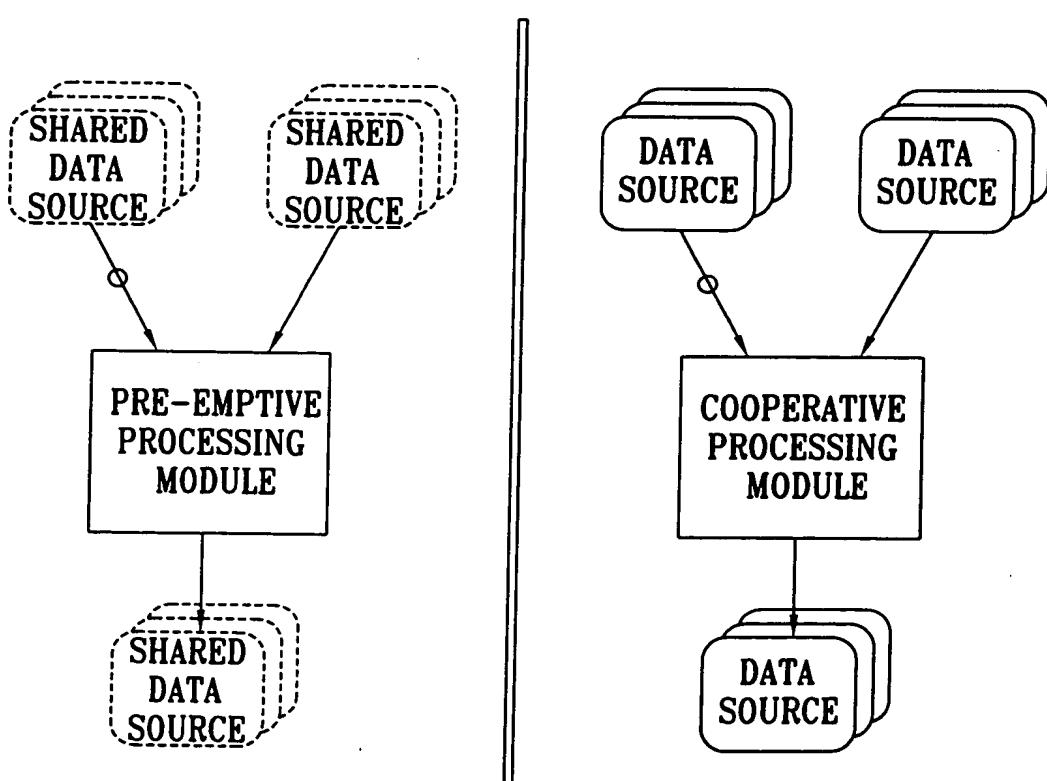
**FIG. 44**

FIG. 45

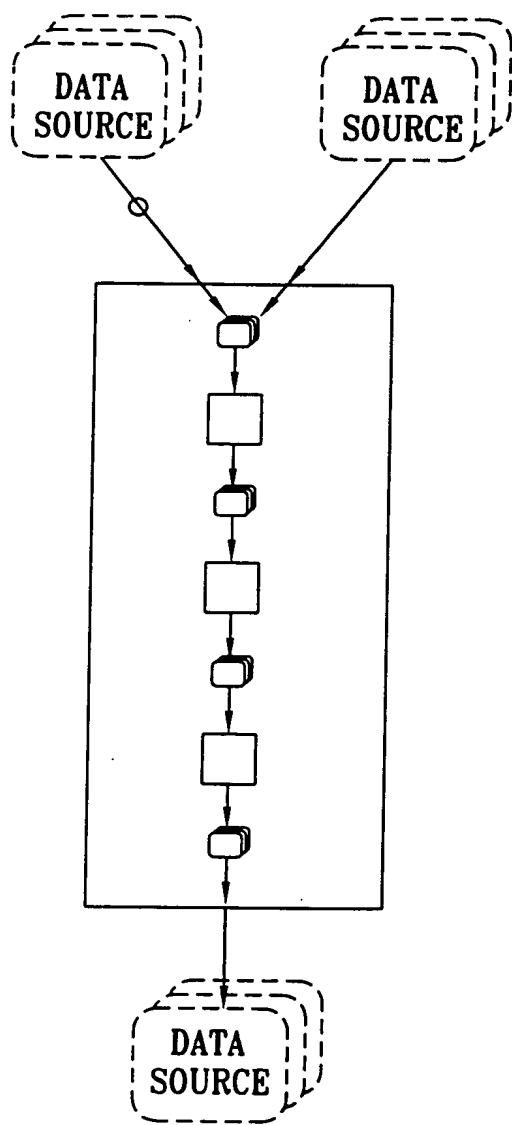
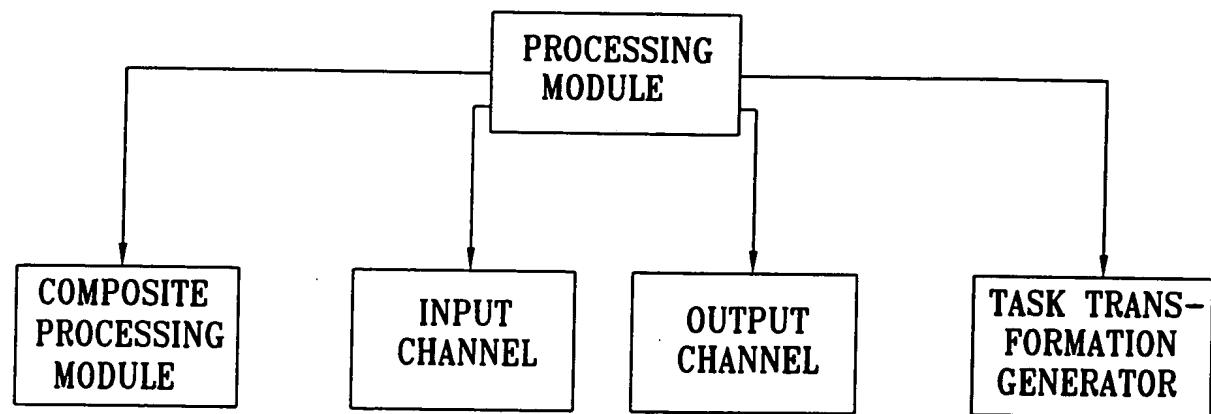


FIG. 46

FIG. 47

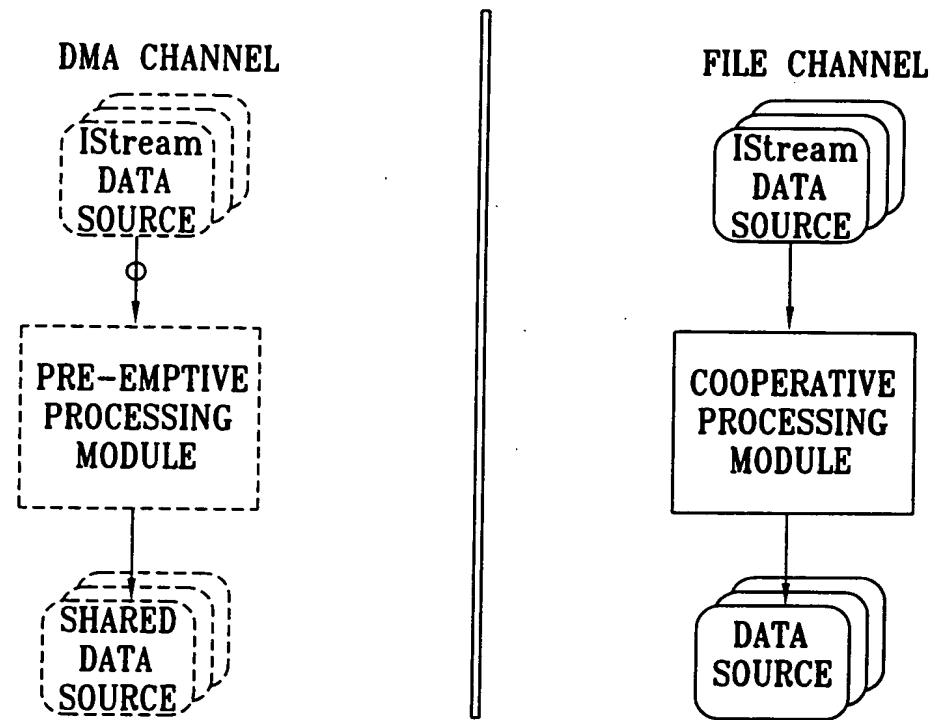


FIG. 48

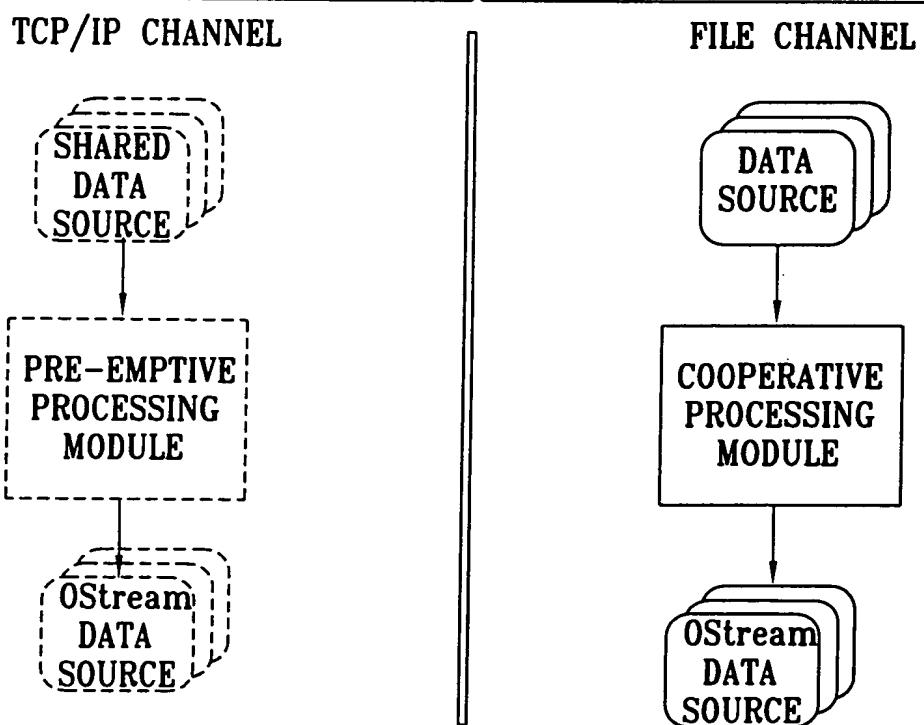


FIG. 49

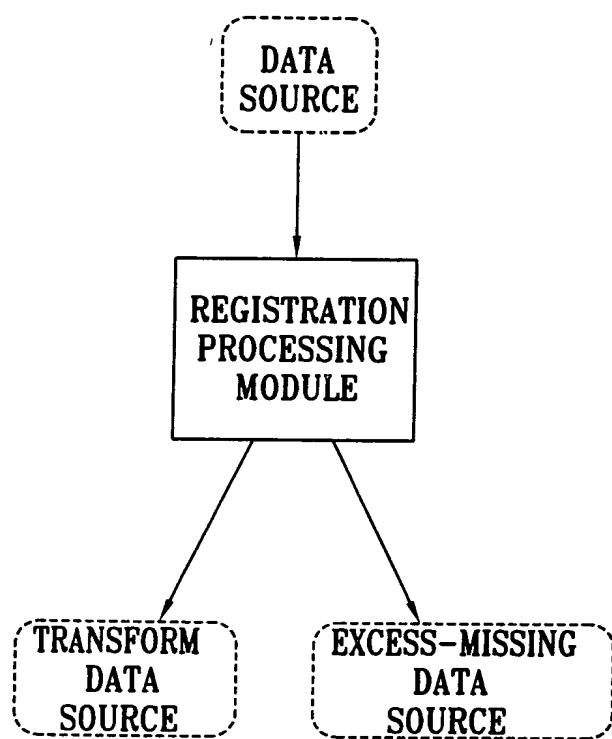


FIG. 50

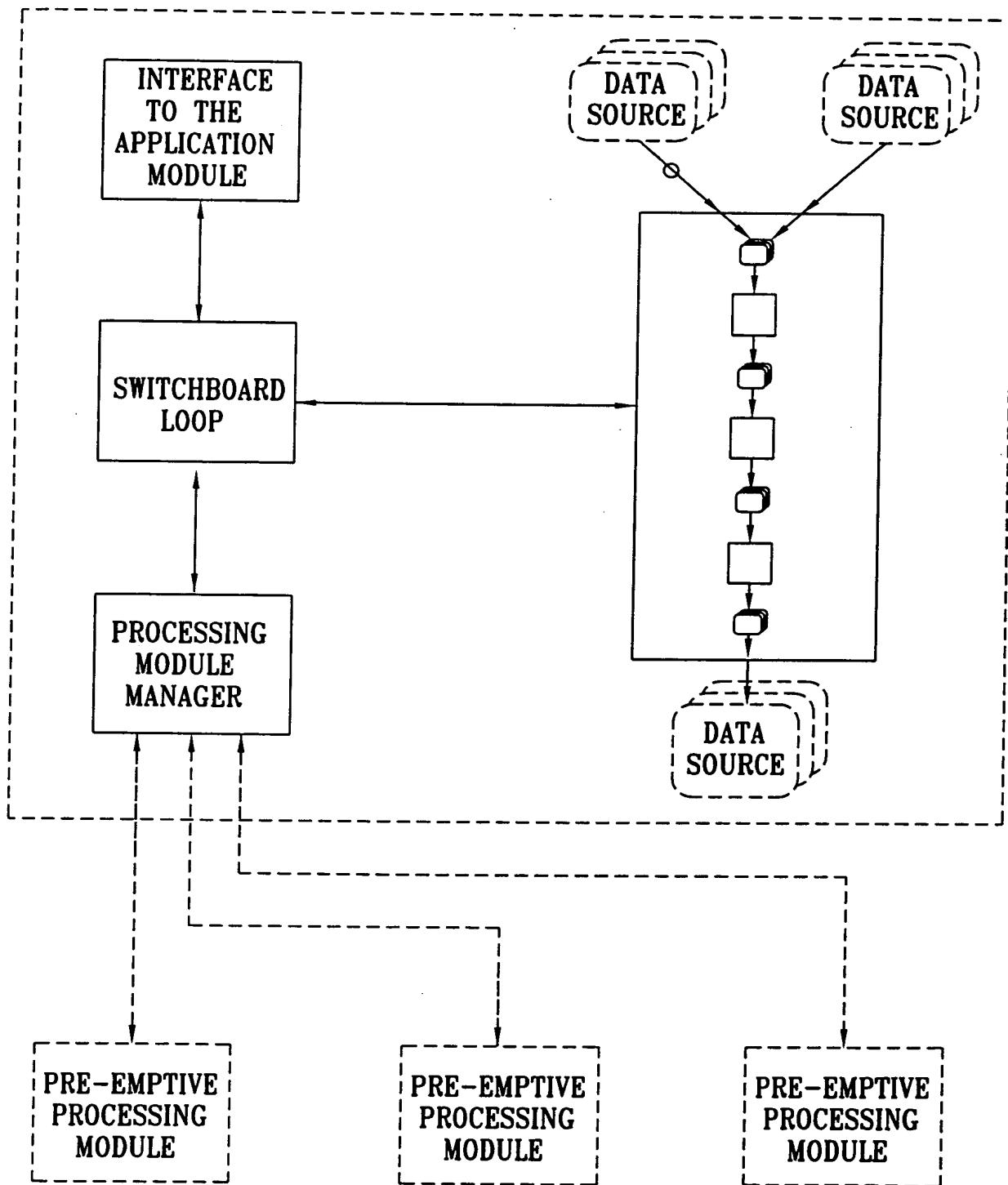


FIG. 51

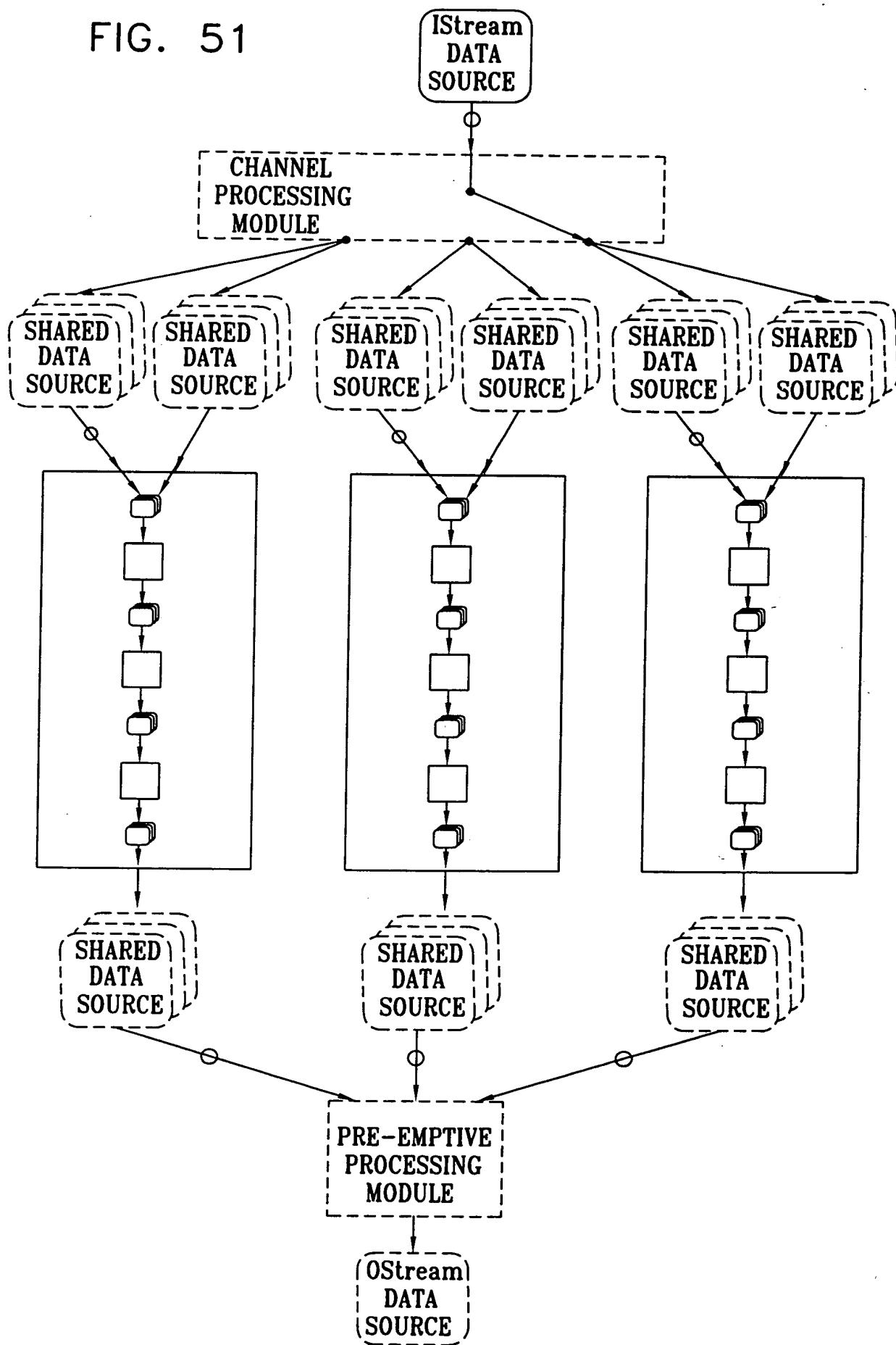


FIG. 52

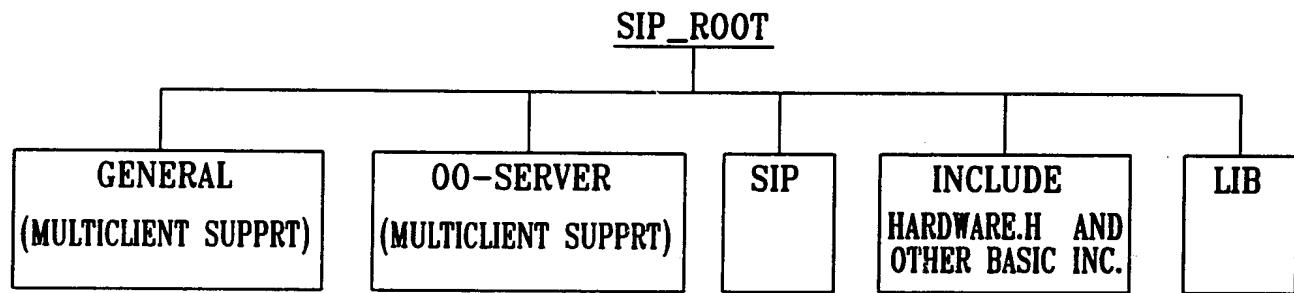


FIG. 53

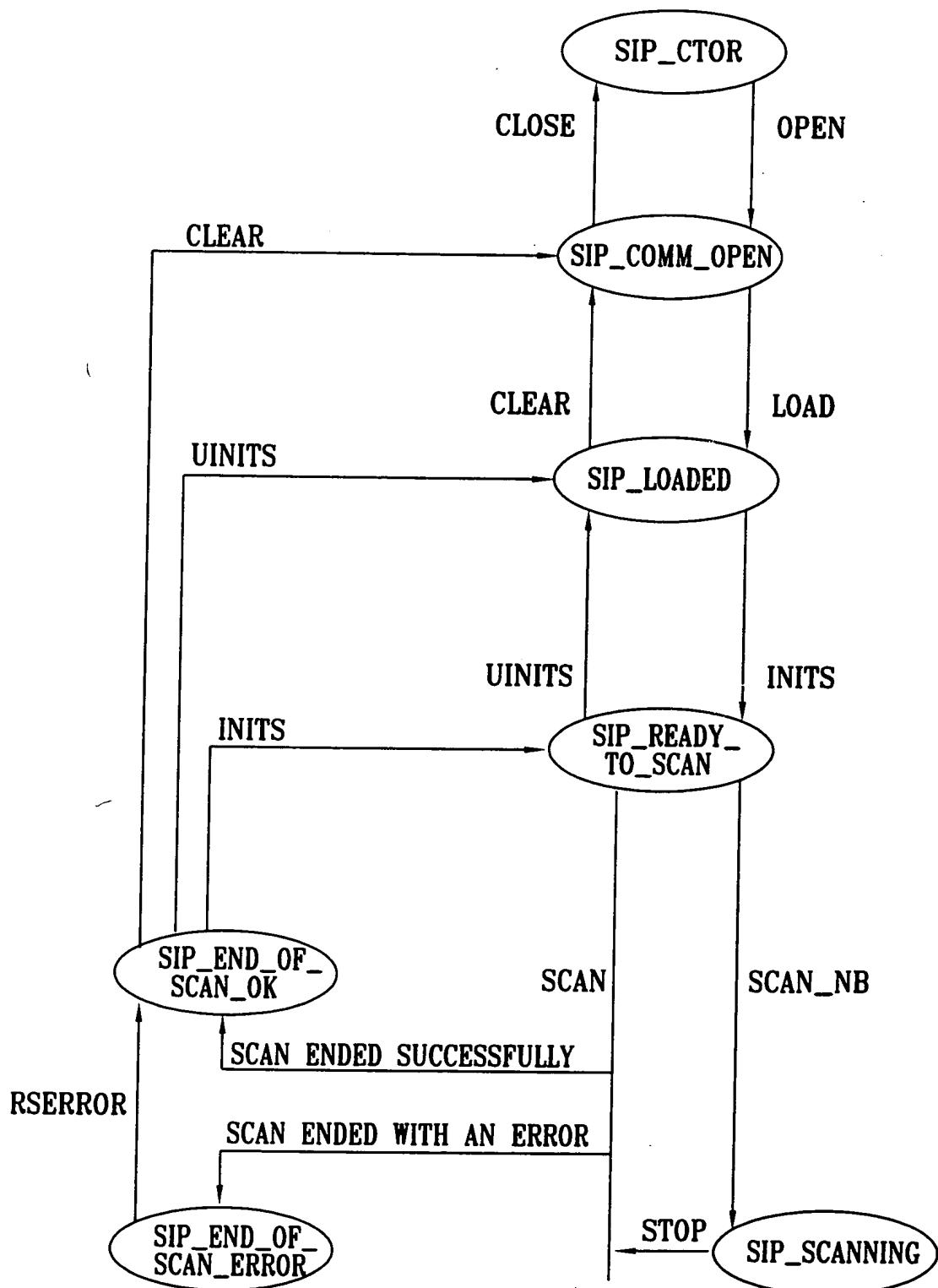
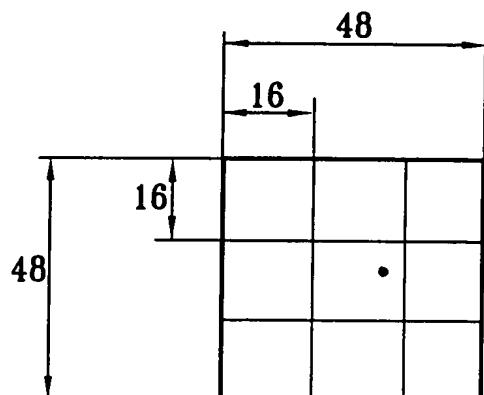
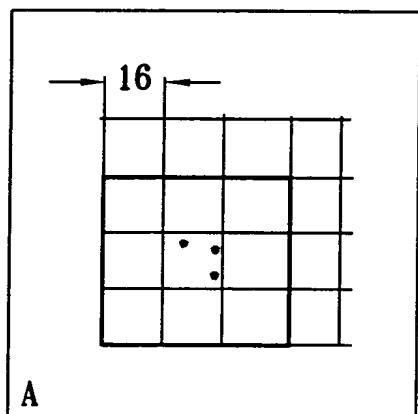


FIG. 54



- DENOTES THE COLOUR DEFECT REPORT DUE TO WHICH THE SNAP OF SIZE 48X48 IS RECORDED.

FIG. 55



- DENOTES A SIGNAL COLOR_DEFECT REPORT
- DENOTES AREA OF THE SNAP.

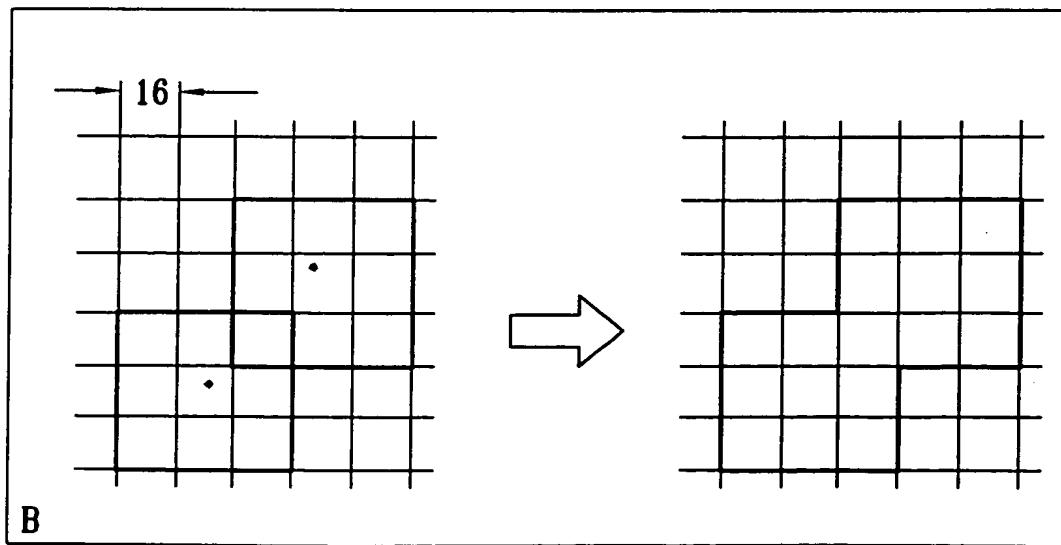


FIG. 56

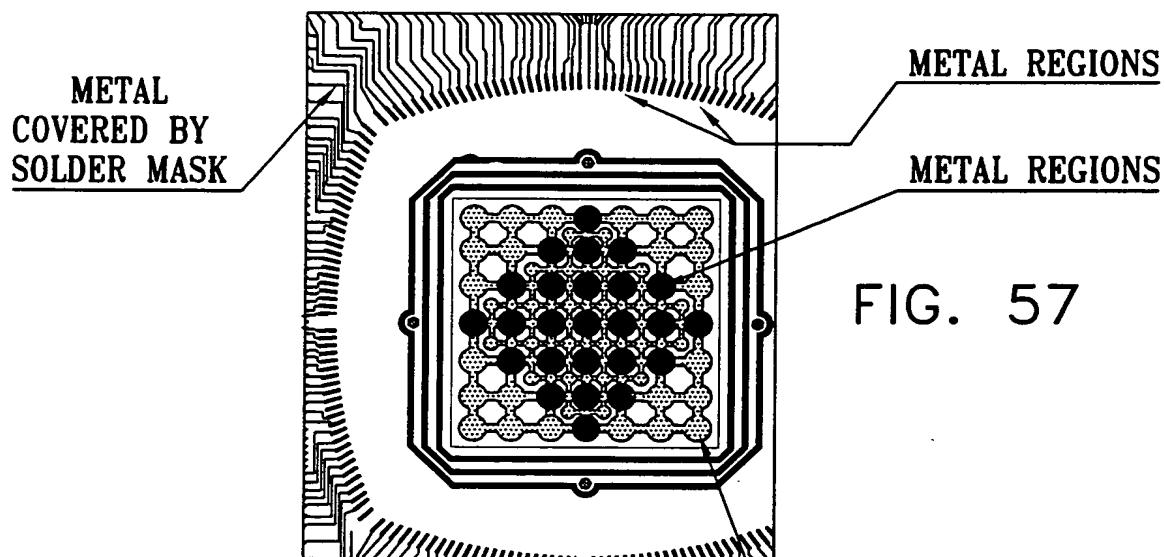
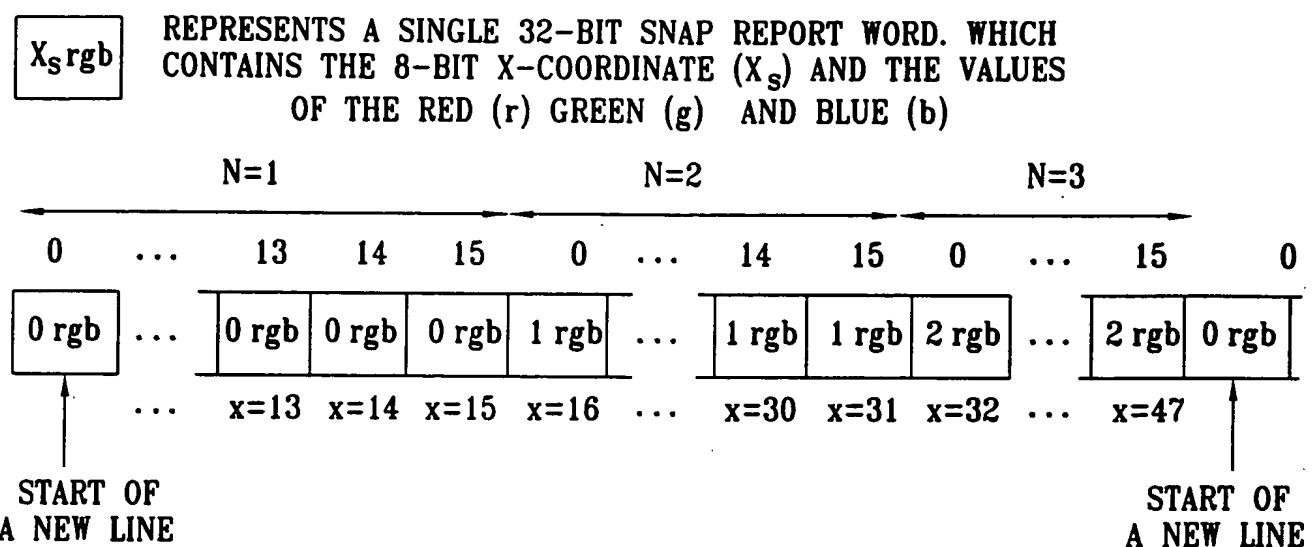


FIG. 58 REGION TO BE IGNORED

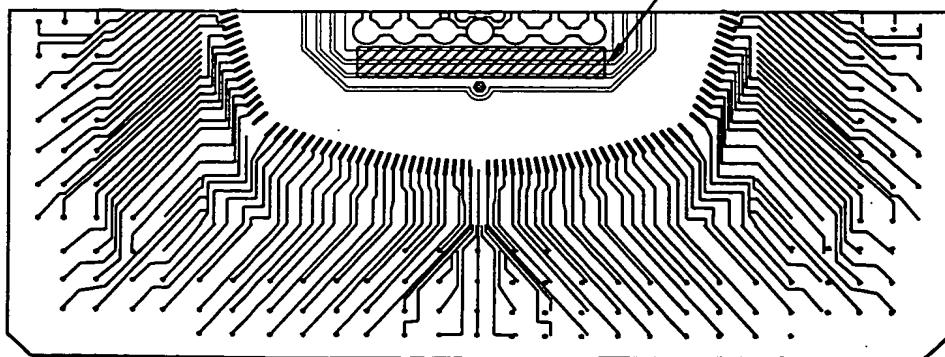


FIG. 59

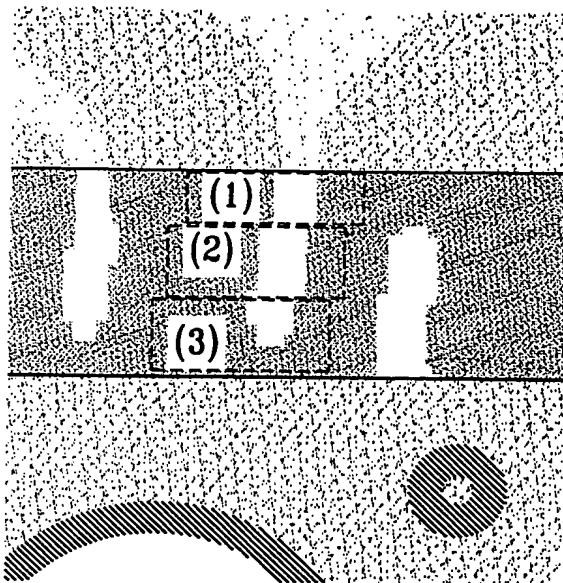


FIG. 60A

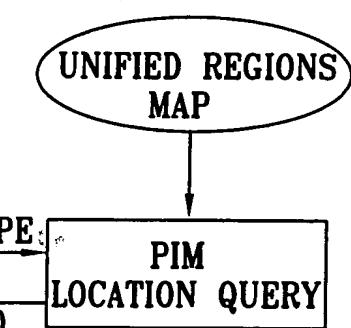
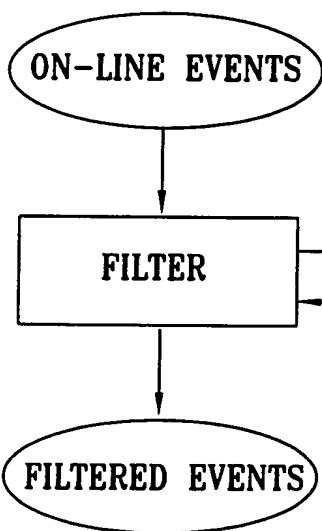
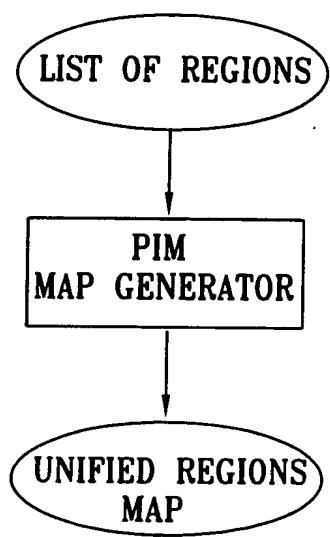


FIG. 60B

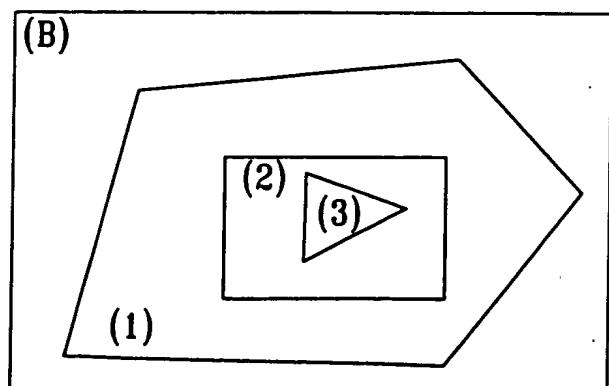
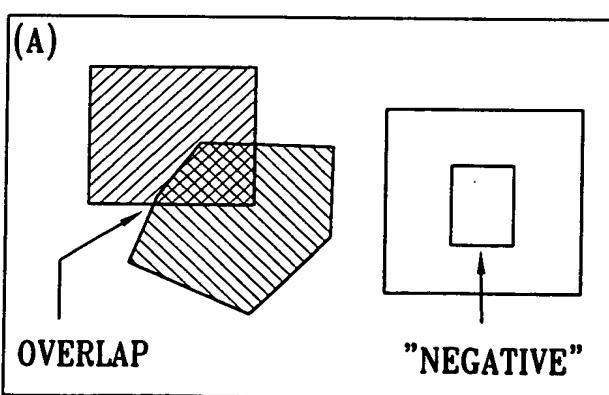


FIG. 61

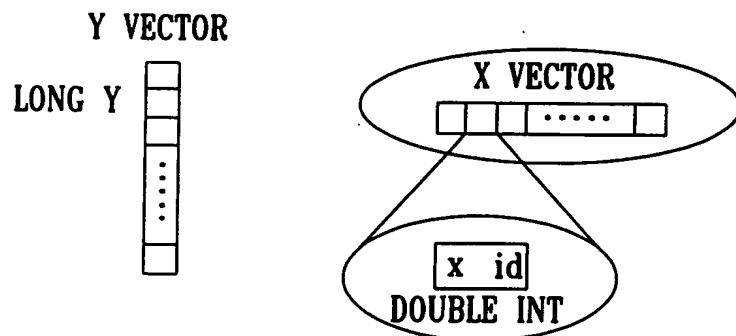


FIG. 62

WHERE THE GRAPHIC INTERPRETATION IS:

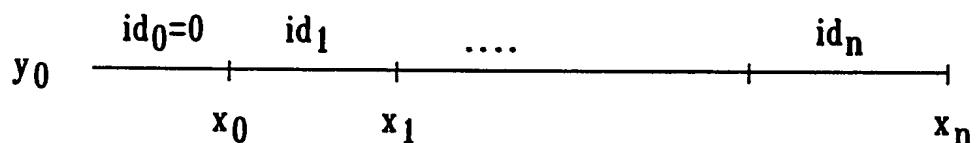


FIG. 63

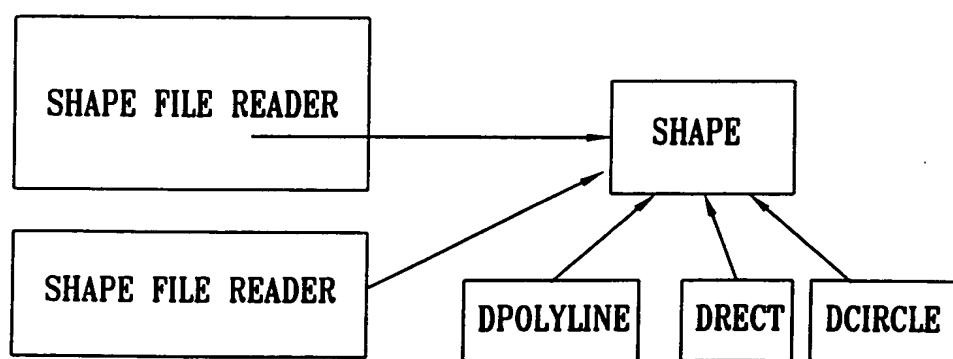
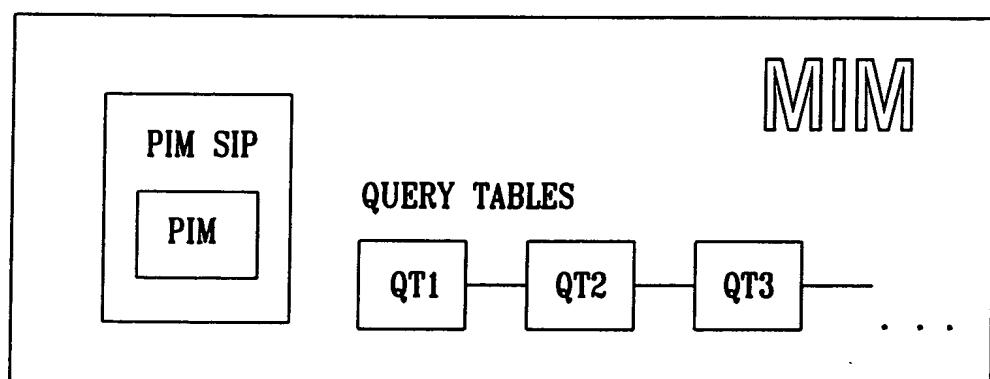


FIG. 64

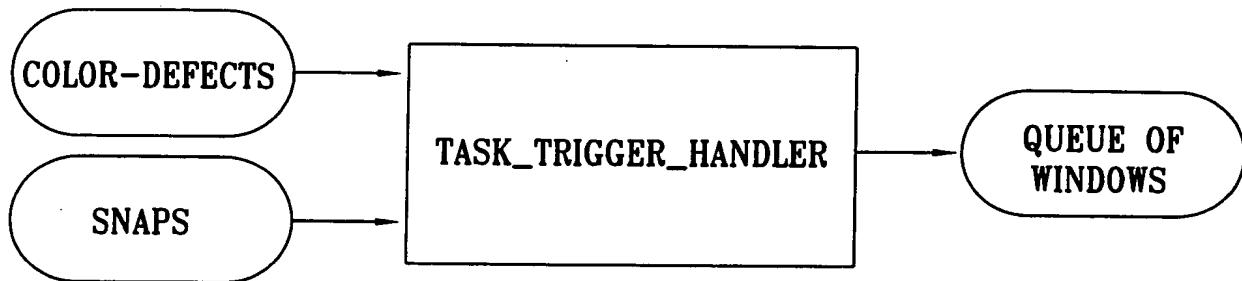


FIG. 65

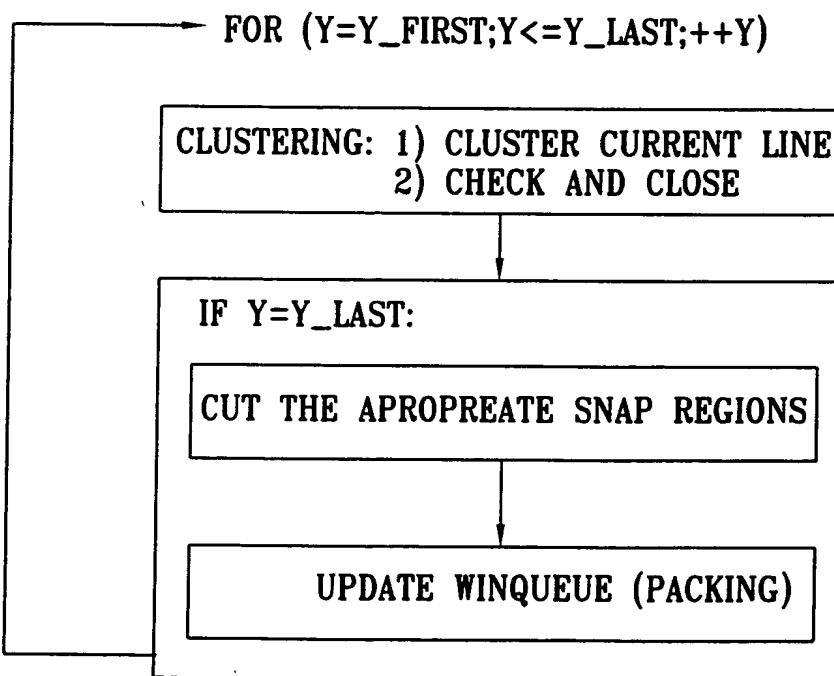
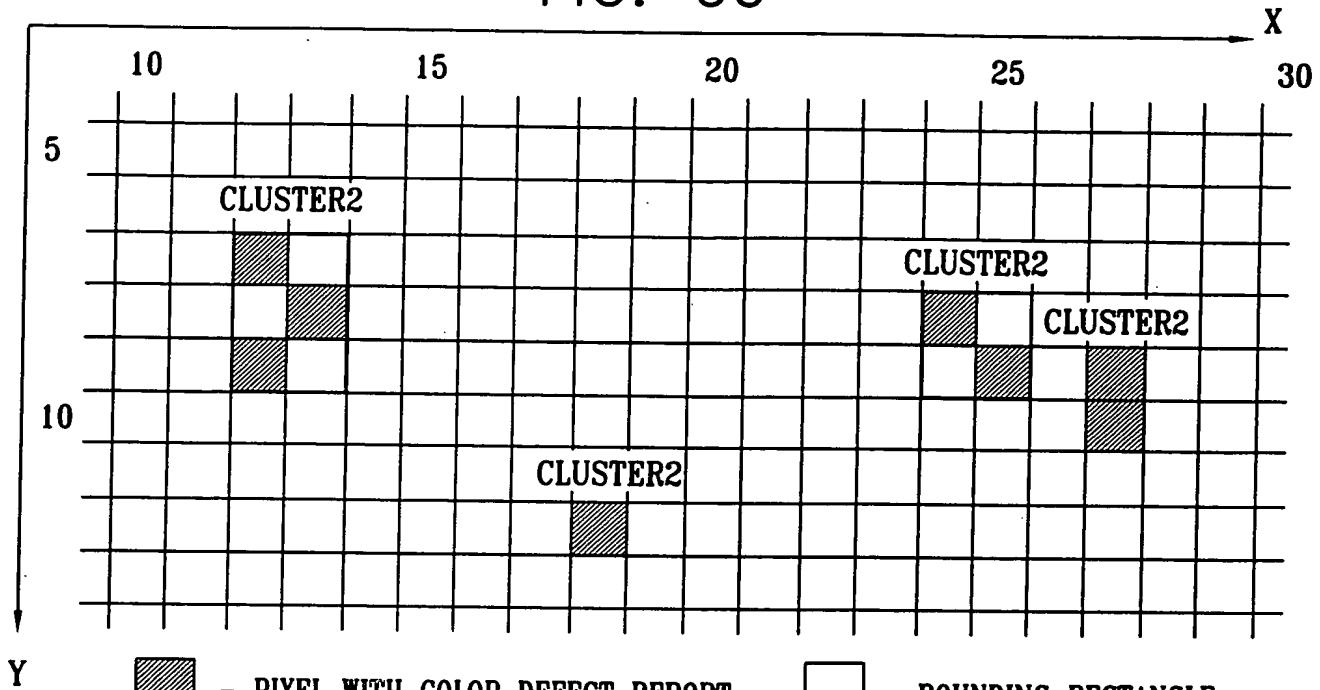


FIG. 66



Y

 - PIXEL WITH COLOR DEFECT REPORT - BOUNDING RECTANGLE

THE RUN LENGTH STRUCTURE FOR THE CASE WHERE THE SNAP REGION IS THE BOUNDING RECTANGLE ENLARGED BY TWO PIXELS ON EACH SIDE:

FIG. 67A

THE SNAP RECTANGULAR REGIONS (COLORED AREAS)
THAT WERE CUT ACCORDING TO THE CLUSTERS

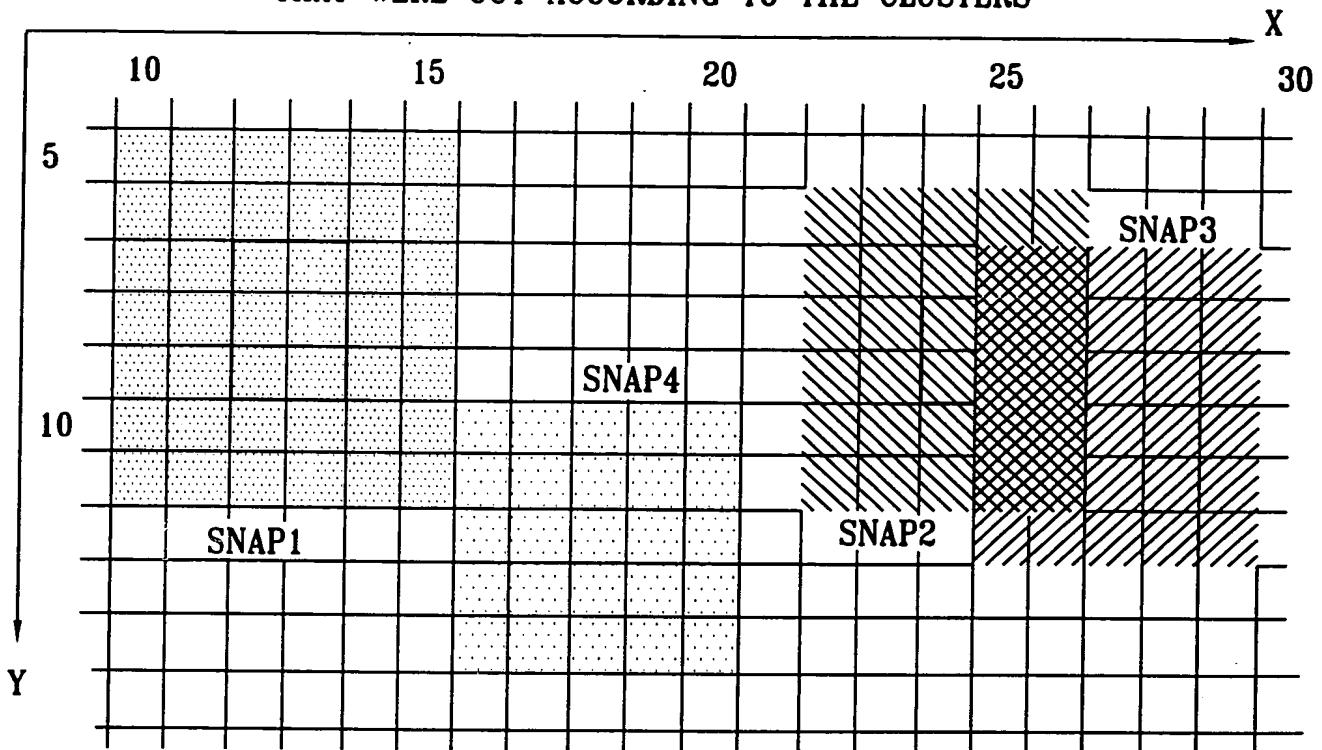


FIG. 67B

THE RESULT IS 4 WINDOWS, EACH POSSESS THE CLUSTER
AND A RECTANGLE WITH THE RED-DATA.

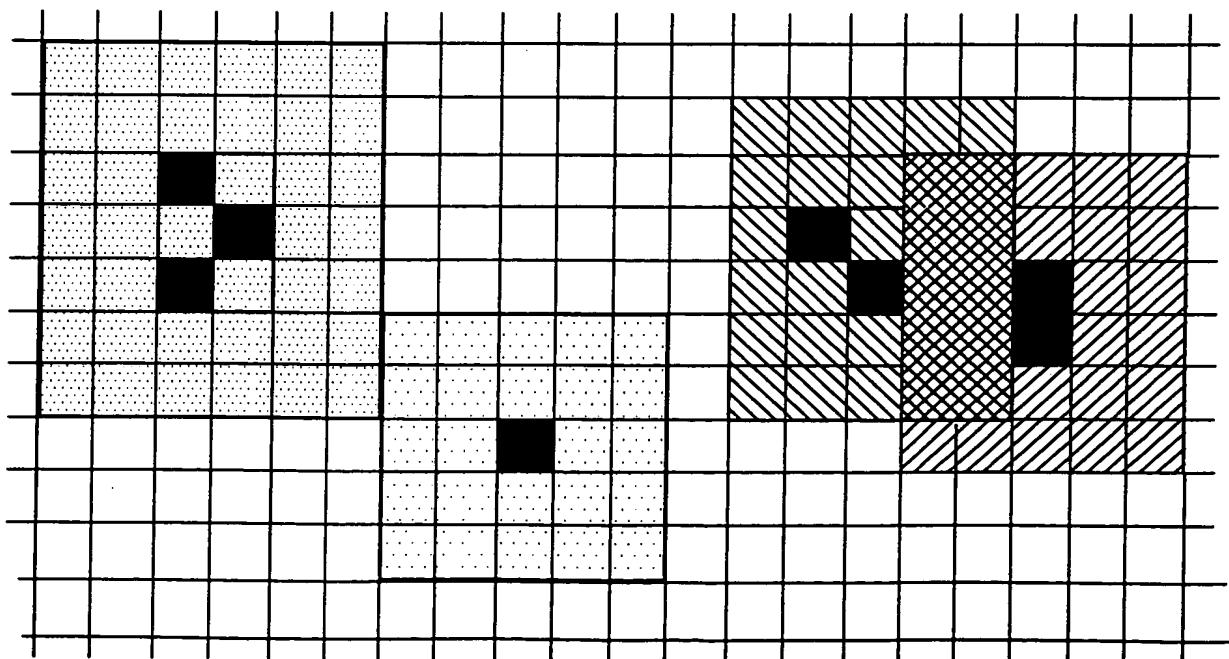


FIG. 68

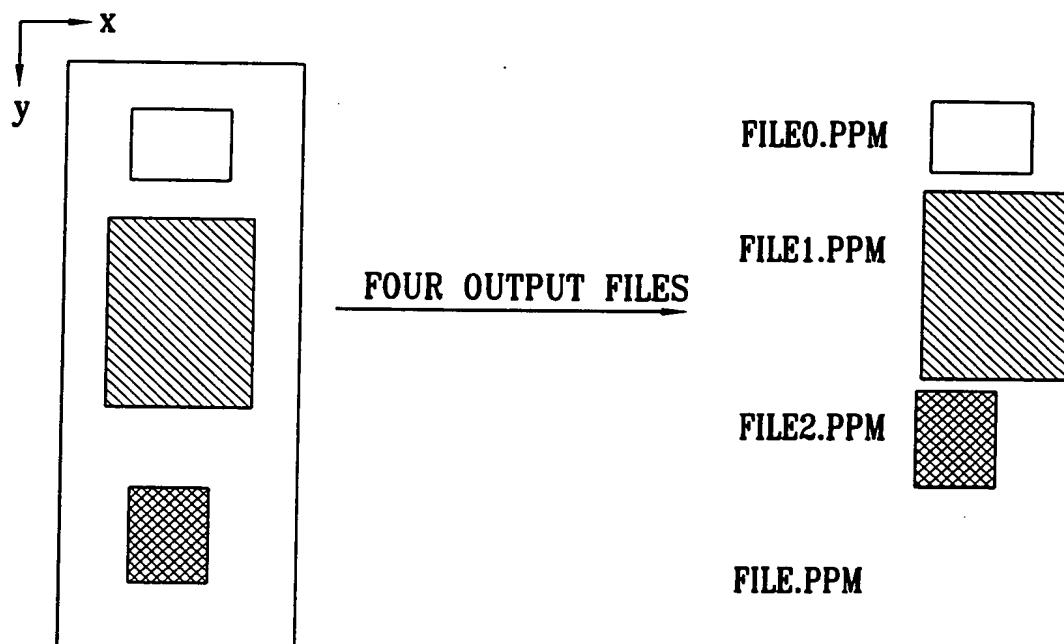


FIG. 69

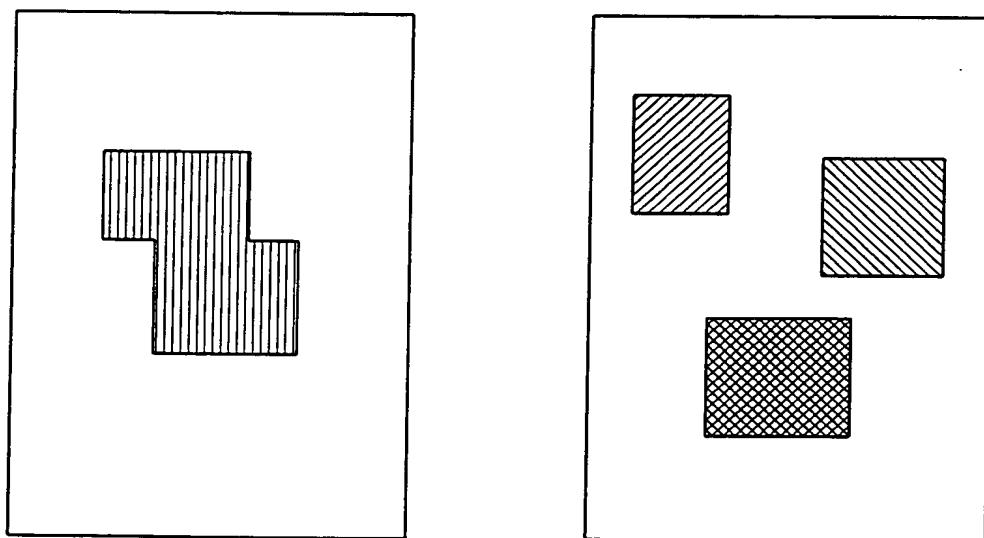


FIG. 70

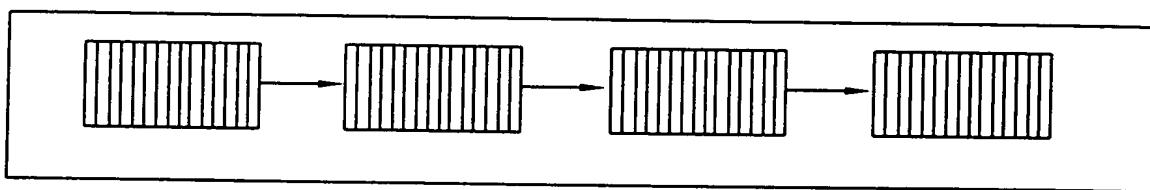


FIG. 71

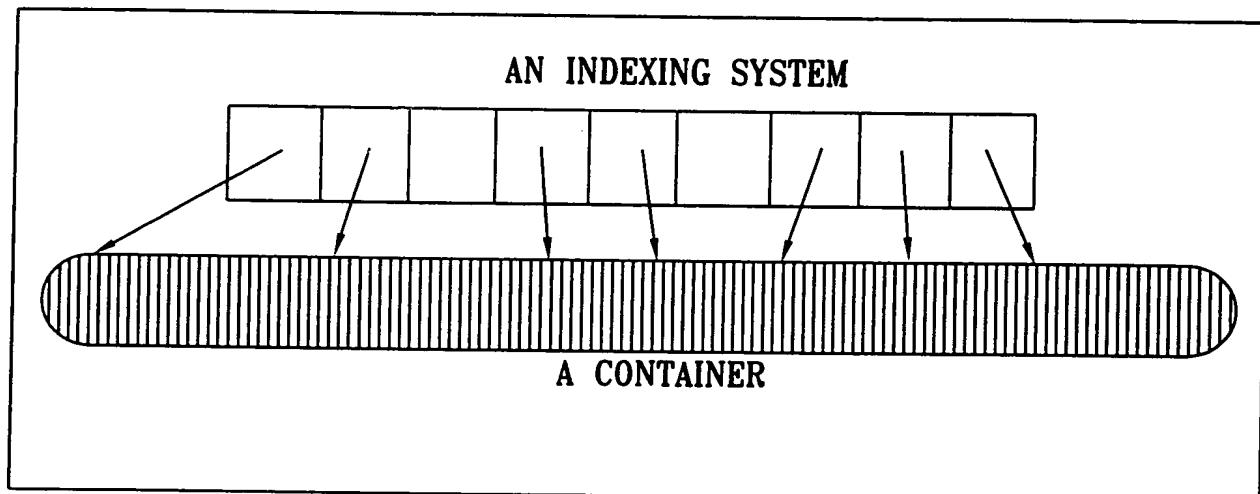


FIG. 72

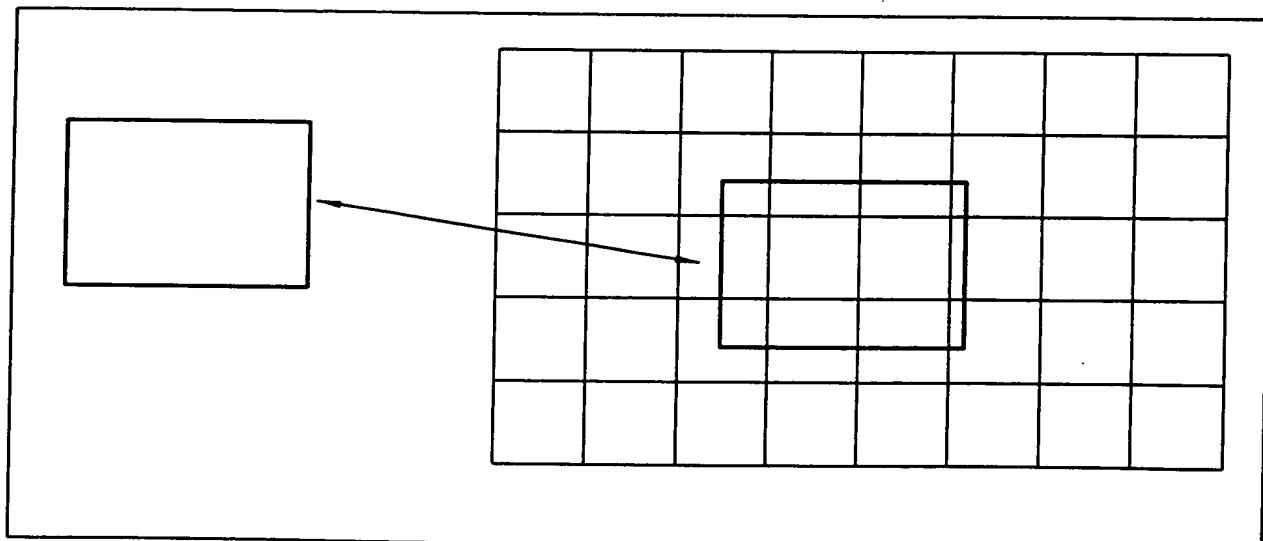


FIG. 73

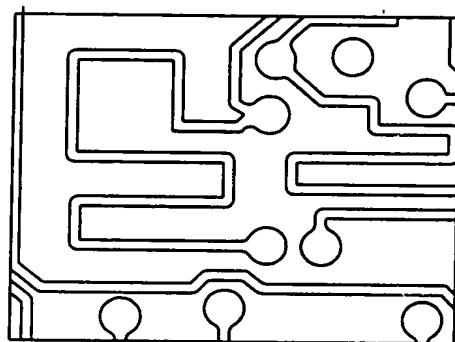


FIG. 74

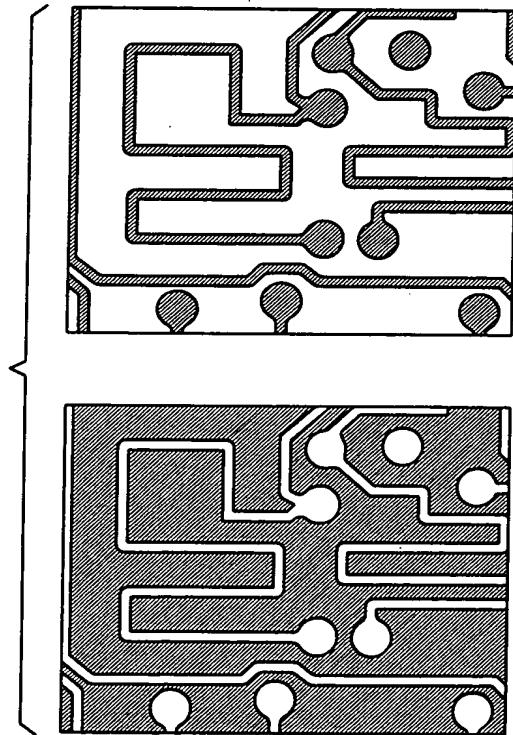
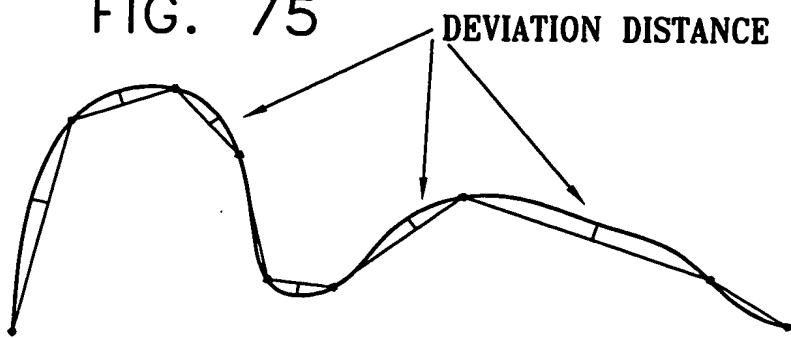
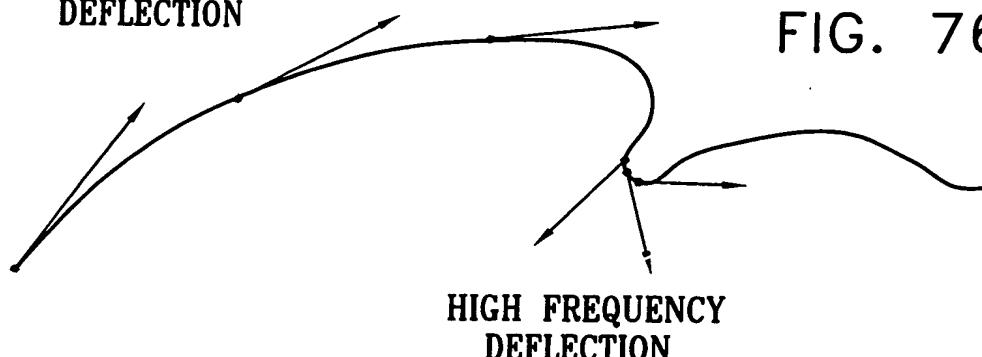


FIG. 75



LOW FREQUENCY DEFLECTION

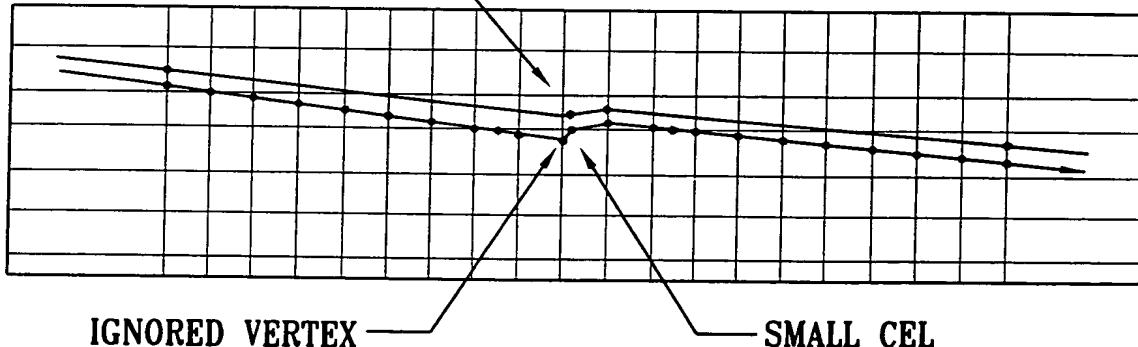
FIG. 76



HIGH FREQUENCY DEFLECTION

RESULTING CONTOUR

FIG. 77



MAXIMUM DISTANCE FROM ORIGINAL CONTOUR

FIG. 78

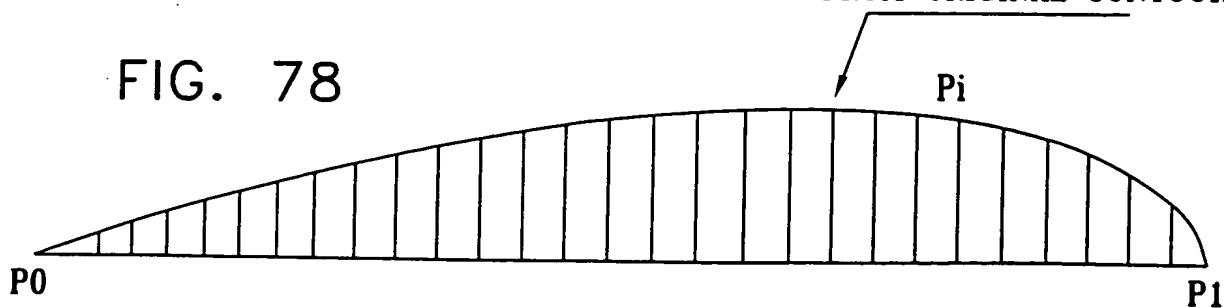


FIG. 79

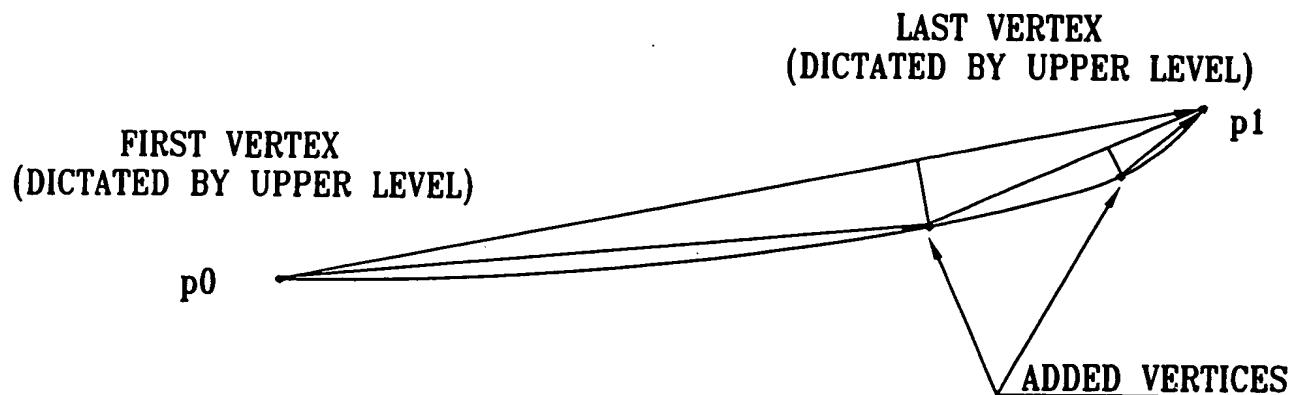


FIG. 80

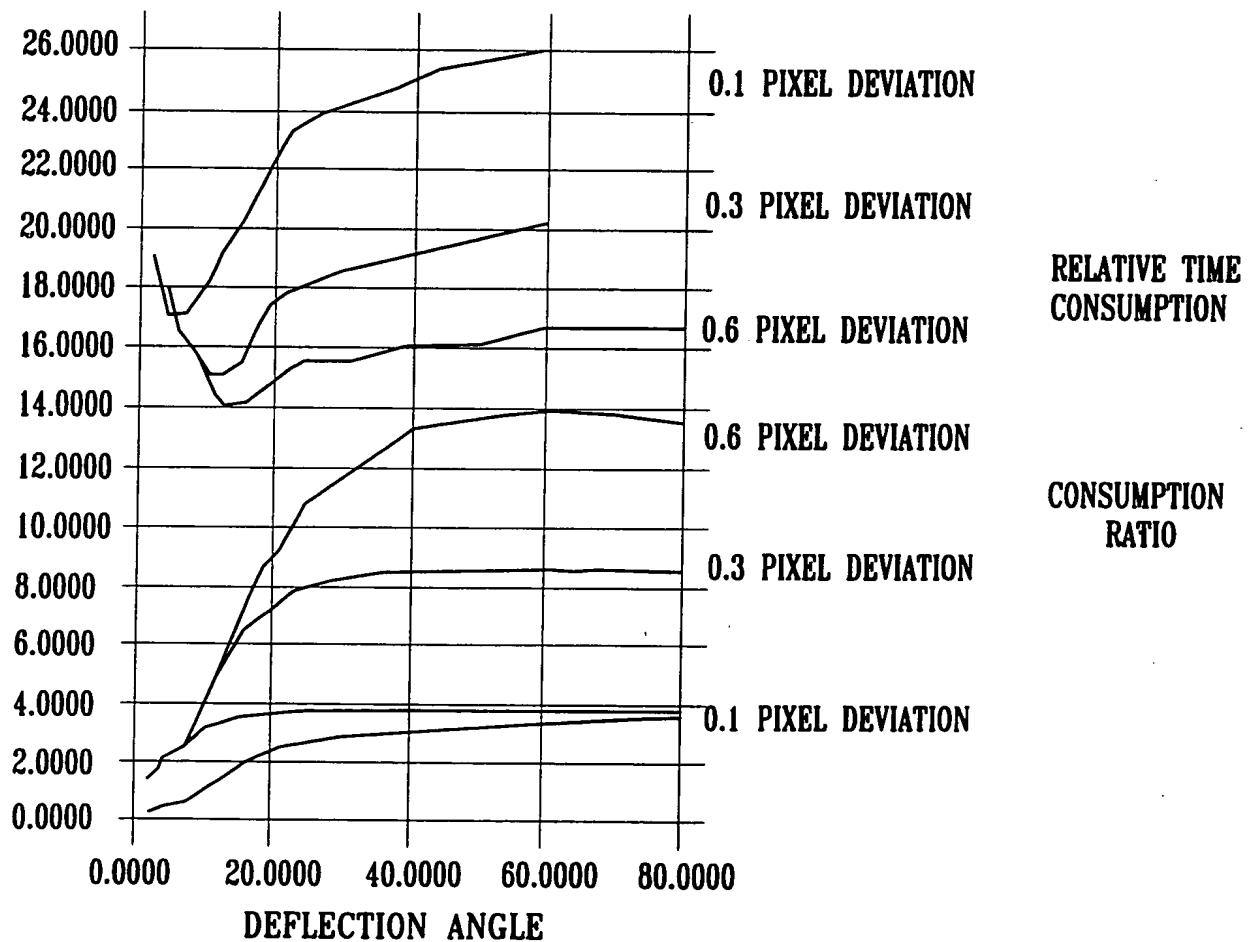


FIG. 81

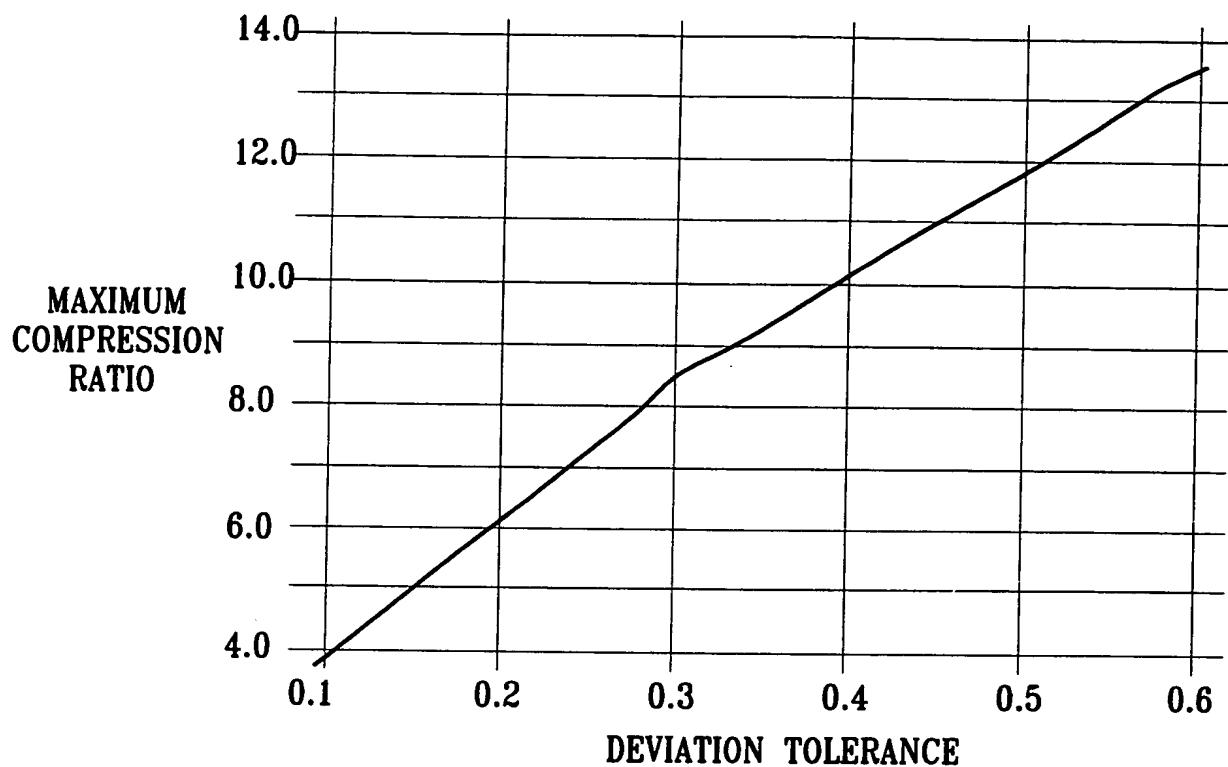


FIG. 82

